

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA CONSULTA DE
BIBLIOGRAFÍA PERTENECIENTE A LA BIBLIOTECA DE LA
UNIVERSIDAD LIBRE SEDE BOSQUE POPULAR**

JULIÁN ENRIQUE OSORIO AMAYA

CODIGO 66041077

UNIVERSIDAD LIBRE

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ

ENERO 2013

**DESARROLLO DE UNA APLICACIÓN MÓVIL PARA CONSULTA DE
BIBLIOGRAFÍA PERTENECIENTE A LA BIBLIOTECA DE LA
UNIVERSIDAD LIBRE SEDE BOSQUE POPULAR**

JULIÁN ENRIQUE OSORIO AMAYA

CODIGO 66041077

**TRABAJO DE GRADO COMO REQUISITO PARA OPTAR POR EL TÍTULO
DE INGENIERO DE SISTEMAS**

DIRECTOR:

Ing. Yaneth Cárdenas Sánchez

UNIVERSIDAD LIBRE

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

BOGOTÁ

ENERO 2013

Nota de aceptación

Jurado

Jurado

Jurado

Jurado

Bogotá, Enero 26 de 2013

A mi familia y amigos por su apoyo incondicional

AGRADECIMIENTOS

La culminación de este proyecto de grado es producto del esfuerzo y la dedicación de las personas que me rodearon durante este proceso que tomó dos años. Quiero agradecer profundamente a mis padres por la paciencia que han tenido durante el tiempo que me ha tomado terminar este proyecto de grado, a la Ingeniera Yaneth Cárdenas que con su experiencia me guió en la forma y los contenidos que debía llevar el documento, a Andrea Ravagli que siempre estuvo a mi lado para animarme y finalmente a mis amigos que siempre estuvieron pendientes.

TABLA DE CONTENIDO

LISTAS ESPECIALES	8
GLOSARIO	12
INTRODUCCIÓN	18
1. PLANTEAMIENTO DEL PROBLEMA.....	19
1.1 FORMULACIÓN DEL PROBLEMA	19
1.2 JUSTIFICACIÓN	20
1.3 OBJETIVOS	22
1.4 ALCANCE.....	23
1.5 HIPÓTESIS	23
1.6 VARIABLES.....	24
1.7 DISEÑO METODOLÓGICO	24
2. MARCO REFERENCIAL	28
2.1 MARCO HISTÓRICO	28
2.2 MARCO TEÓRICO.....	30
2.3 MARCO CONCEPTUAL.....	32
2.4 MARCO LEGAL.....	33
2.5 MARCO TECNOLÓGICO.....	33
3. ESTRUCTURA TEMÁTICA	39
3.1 INGENIERÍA DEL PROYECTO.....	39
4. RESULTADOS Y DISCUSIÓN	136
4.1 MODELO DE NEGOCIO	136
4.2 CASOS DE USO	136
4.3 ARQUITECTURA DEL SISTEMA.....	137

4.4	NAVEGACIÓN DEL SISTEMA.....	142
5.	CONCLUSIONES	143
6.	RECOMENDACIONES.....	145

LISTAS ESPECIALES

LISTA DE TABLAS

Tabla 1 - Variables.....	24
Tabla 2 - Riesgos en la fase de inicio	40
Tabla 3 - Caso de uso para búsqueda simple de material.....	47
Tabla 4 - Caso de uso para la consulta avanzada de material	50
Tabla 5 - Caso de uso para la reserva de material	54
Tabla 6 - Requerimientos del Sistema Actual	57
Tabla 7 - Especificaciones del sistema propuesto	61
Tabla 8 - Riesgos en la fase de elaboración.....	66
Tabla 9 - Caso de uso para la consulta de bibliografía por título	69
Tabla 10 - Caso de uso para la consulta de bibliografía por autor.....	71
Tabla 11 - Caso de uso para la consulta de bibliografía por tema.....	74
Tabla 12 - Caso de uso para consulta avanzada de bibliografía	76
Tabla 13 - Caso de uso para el login de usuario.....	79
Tabla 14 - Caso de uso para la validación del rol de usuario	81
Tabla 15 - Caso de uso para la transformación del resultado.....	83
Tabla 16 - Caso de uso para la validación del resultado de la consulta	85
Tabla 17 - Riesgos en la fase de construcción	97
Tabla 18 – Diccionario de datos para la tabla Libros	106
Tabla 19 – Diccionario de datos para la tabla Usuarios.....	107
Tabla 20 – Diccionario de datos para la tabla Roles.....	108
Tabla 21 - Diccionario de datos para la tabla Roles Usuarios	109
Tabla 22 - Riesgos en la fase de transición	131
Tabla 23 - Técnicas para las pruebas de software orientado a objetos	132

LISTA DE FIGURAS

Ilustración 1 - Estructura de la tecnología JME	34
Ilustración 2 - Especificación de la tecnología JEE	35
Ilustración 3 - Logo de Visual Paradigm	35
Ilustración 4 - Logo del ambiente de desarrollo integrado Netbeans	36
Ilustración 5 - Arquitectura del motor de base de datos MySQL	37
Ilustración 6 - Arquitectura de un servicio web	38
Ilustración 7 - Arquitectura cliente-servidor del sistema actual	42
Ilustración 8 - Modelo de negocio del sistema actual.....	43
Ilustración 9 - Diagrama de casos de uso del sistema actual	45
Ilustración 10 - Diagrama de casos de uso del sistema propuesto	60
Ilustración 11 - Modelo de negocio del sistema propuesto	68
Ilustración 12 - Diagrama de actividad para el ingreso a la aplicación.....	88
Ilustración 13 - Diagrama de actividad para la consulta de bibliografía por título	89
Ilustración 14 - Diagrama de actividad para la consulta de bibliografía por autor.....	90
Ilustración 15 - Diagrama de actividad para la consulta de bibliografía por tema.....	91
Ilustración 16 - Diagrama de actividad para la consulta avanzada de bibliografía	92
Ilustración 17 - Diagrama de secuencia para el ingreso del usuario a la aplicación móvil.....	94
Ilustración 18 - Diagrama de secuencia para la consulta de material	96
Ilustración 19 - Arquitectura de la solución propuesta	99
Ilustración 20 - Proceso de generación del esquema	101

Ilustración 21 - Comunicación entre el servicio web y el Enterprise Java Bean	102
Ilustración 22 - Ciclo de vida de una aplicación móvil con la tecnología Java Micro Edition	103
Ilustración 23 - Arquitectura de la Aplicación Móvil	104
Ilustración 24 - Modelo de base de datos	105
Ilustración 25 - Diagrama de clases para la aplicación móvil	110
Ilustración 26 - Diagrama de clases para la lógica de negocio	119
Ilustración 27 - Diagrama de clases para la aplicación web	125
Ilustración 28 - Diagrama de componentes	129

LISTA DE ANEXOS

Anexo A. Entrevista realizada a Liliana López, directora de la biblioteca de la Universidad Libre sede Bosque Popular.

Anexo B. Formato de la encuesta realizada a estudiantes y docentes de la Universidad Libre sede Bosque Popular.

Anexo C. Resultados de la encuesta realizada a estudiantes y docentes de la Universidad Libre sede Bosque Popular

Anexo D. Disco compacto con el código fuente desarrollado para el proyecto.

Anexo E. Manual de usuario

Anexo F. Deck de pruebas aplicado al software desarrollado para el proyecto.

Anexo G. Scripts de base de datos

GLOSARIO

C

CDC: (Connected Device Configuration) es una configuración para asistentes personales de última generación o teléfonos inteligentes avanzados. Requiere una máquina virtual java compatible con J2SE 1.4.2 o superior.

CLDC: (Connected Limited Device Configuration) es una configuración para dispositivos con potencia de cálculo, memoria, batería y acceso a red limitados que incluye una máquina virtual Java reducida (KVM).

E

EJB: (Enterprise JavaBeans) es un interface de programación que define una arquitectura de componentes para realizar sistemas cliente/servidor multicapa.

J

JAVA: es un lenguaje de programación orientado a objetos. La programación es compilada en bytecode, el cual es ejecutado por la máquina virtual Java. Se usa usualmente el compilador JIT.

JDK: (Herramientas de desarrollo de Java) es un conjunto de programas que hacen posible el desarrollo de aplicaciones Java y con su instalación viene la máquina virtual, que hace posible la interpretación del bytecode.

JAX-WS: es una tecnología para la construcción de servicios web y los clientes que se comunican a través de XML. Permite a los desarrolladores escribir mensajes orientados a RPC (Remote Call Procedure), así como orientado a servicios web.

JEE: (Java Enterprise Edition) es un ambiente de Sun enfocado en Java, independiente de la plataforma, para el desarrollo, construcción e implementación de aplicaciones online para empresas basadas en Web.

JME: (Java Micro Edition) permite que los desarrolladores usen Java y las herramientas inalámbricas JME para crear aplicaciones y programas para dispositivos móviles inalámbricos. Consiste de dos elementos: configuraciones y perfiles. Las configuraciones proveen las librerías y una máquina virtual para una categoría de dispositivos inalámbricos. Los perfiles son APIs integradas por encima de las configuraciones para proveer un ambiente de funcionamiento para dispositivos específicos, como PDAs, telefonía celular o set-top boxes.

JPA: Proporciona un modelo de persistencia basado en POJOs para mapear las bases de datos relacionales en Java.

M

MIDlet: es una aplicación desarrollada utilizando la plataforma J2ME y construida sobre la configuración CLDC utilizando el perfil MIDP. Los MIDlets son aplicaciones que se desarrollan para los teléfonos móviles actuales con soporte para CLDC 1.0 ó 1.1 y MIDP 1.0 ó 2.0.

MIDP: es perfil para dispositivos de información móviles (Mobile Information Device Profile, MIDP) y se combina con la configuración CLDC para proporcionar un entorno de ejecución para dispositivos móviles (teléfonos, PDAs, etc). Los usuarios pueden seleccionar las aplicaciones MIDP situadas en un servidor web.

MySQL: es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL es software libre.

P

PROTOTIPO: es un modelo del comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos. Es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas.

R

RUP: es el proceso Unificado de Rational (Rational Unified Process). Es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP no es un sistema con pasos firmemente establecidos sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

S

SERVICIOS WEB: (Web service) es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones de software desarrolladas en lenguajes de programaciones diferentes y ejecutadas sobre cualquier plataforma. Pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

SERVIDOR DE APLICACIONES: es un servidor en una red de computadores que ejecuta ciertas aplicaciones. Usualmente se trata de un dispositivo de software que proporciona servicios de aplicación a las computadoras cliente. Un servidor de aplicaciones gestiona la mayor parte o la totalidad de las funciones de lógica de negocio y de acceso a los datos de

la aplicación. Ayudan a la centralización y disminución de la complejidad en el desarrollo de aplicaciones.

SOAP: proporciona un mecanismo simple y ligero de intercambio de información entre dos puntos usando el lenguaje XML. SOAP no es más que un mecanismo sencillo de expresar la información mediante un modelo de empaquetado de datos modular y una serie de mecanismos de codificación de datos.

SOA: es el acrónimo para la arquitectura orientada a servicios, la cual está compuesta por diferentes estándares como SOAP y WSDL que facilitan la publicación de procesos de negocio que son construidos como componentes de software para que puedan ser reutilizados por otras aplicaciones.

U

UML: (Unified Modeling Language - Lenguaje Unificado de Modelado) es un lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Se utiliza para definir un sistema de software, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc.

W

Wi Fi: (Wireless Fidelity) es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11 creado para redes locales inalámbricas, pero que también se utiliza para acceso a Internet.

X

XML: (Extensible Markup Language) es una especificación/lenguaje de programación desarrollada por el W3C. XML es una versión de SGML,

diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.

RESUMEN

El presente proyecto tiene como objetivo poner al alcance de los usuarios una aplicación diseñada para dispositivos móviles que permitirá a la comunidad académica de la Universidad Libre sede Bosque Popular realizar consultas de bibliografía. Se implementará la metodología RUP que se caracteriza por el uso de UML y la producción de prototipos como resultado de las iteraciones en sus fases que corresponden al ciclo de vida del producto. Del lado de las tecnologías a usar tenemos JME empleada para el desarrollo de la aplicación móvil y JEE para la lógica del negocio en el lado servidor compuesta por un servicio web y EJB.

Como resultado del proceso aplicado en este proyecto se obtienen una aplicación móvil y un componente web que ponen al alcance de los usuarios la información que posee la biblioteca en sus dispositivos móviles. Se espera que el desarrollo de aplicaciones móviles y empresariales tenga un espacio cada vez más importante dentro del programa de Ingeniería de Sistemas.

PALABRAS CLAVES

JME, JEE, RUP, Ingeniería de software, Aplicaciones móviles

INTRODUCCIÓN

Las tecnologías móviles están adquiriendo una gran importancia en el campo del desarrollo de software ya que en el mercado actual podemos encontrar una gran variedad de dispositivos compuestos por sistemas operativos y hardware que ofrecen al usuario una experiencia mejorada orientada hacia Internet con respecto a los dispositivos que se ofrecían en años anteriores.

Los motivos que definieron el desarrollo de una aplicación móvil son la posibilidad de brindar al usuario la facilidad de realizar consultas bibliográficas desde un dispositivo diferente a un computador portátil o de escritorio y complementar la aplicación existente. Por otro lado, es una oportunidad para aplicar algunos conocimientos adquiridos en la academia y aprender nuevos, adicionalmente es un campo de acción en pleno crecimiento gracias al aumento en la comunidad de desarrolladores y al interés de grandes empresas como Google, Apple y Nokia en ofrecer a los usuarios una variedad de aplicaciones.

En este proyecto se utilizan metodologías y tecnologías orientadas al desarrollo de una aplicación móvil que saca provecho de la capacidad de conexión a redes inalámbricas de los dispositivos actuales para realizar consultas de bibliografía, de forma que el saldo que tenga el usuario en su cuenta no se verá afectado durante el uso de la aplicación.

1. PLANTEAMIENTO DEL PROBLEMA

En la Universidad Libre Sede Bosque Popular se está presentando una disminución en el uso del Sistema de Información de Consultas de la Biblioteca debido a que los Estudiantes no tienen el tiempo suficiente para trasladarse hasta las terminales de la Biblioteca o a las Salas de Computo para realizar consultas, además manifiestan no tener el interés suficiente por utilizarlo debido a que consultan a través de otros medios como buscadores en Internet.

Por otra parte algunos no tienen conocimiento de la existencia de este Sistema y como resultado no entienden el funcionamiento del mismo. También se detecta una falencia en cuanto a una inducción adecuada frente al uso del Sistema ya que los estudiantes expresan dificultad al usarlo y comentan que no es amigable frente a la consulta de libros o artículos porque requiere varios pasos previos para generar el resultado, así mismo la información no siempre se encuentra actualizada. Igualmente se expone la falta de equipos tanto en la Biblioteca como en las Salas de Computo así como la disponibilidad de los mismos.

En consecuencia a lo anterior se observa una inconformidad por parte de los estudiantes frente al Sistema con el que cuenta actualmente la Biblioteca y por eso se presenta la oportunidad de desarrollar una aplicación móvil que complemente y facilite las consultas de bibliografía.

1.1 FORMULACIÓN DEL PROBLEMA

¿Cómo incentivar a la comunidad académica a consultar el material bibliográfico de la biblioteca de la Universidad Libre Sede Bosque Popular por medio de una aplicación móvil?

1.2 JUSTIFICACIÓN

En la justificación se pretende demostrar desde los puntos de vista técnico, tecnológico, metodológico, económico, social, profesional, investigativo y legal las razones que motivaron el desarrollo de este proyecto. A continuación se describen cada una de las justificaciones:

- **Justificación técnica:**

Es importante conocer el framework CLDC (Connected Limited Device Configuration) que define un conjunto de librerías básicas y las características de la máquina virtual además del perfil MIDP (Mobile Information Device Profile) de los dispositivos móviles que se ofrecen actualmente en el mercado, ya que a partir de estos se determina la configuración y perfil que tienen en común los dispositivos para asegurar el correcto funcionamiento de la aplicación.

- **Justificación tecnológica:**

Se utilizará el protocolo SOAP (Simple Object Access Protocol) para el intercambio de mensajes entre la aplicación móvil y la lógica del negocio desplegada en el servidor de aplicaciones. Además se deben tener conocimientos básicos de la tecnología Java JME como el desarrollo e implementación de las clases MIDlet y clientes de servicios web, que permitirán la construcción de una aplicación móvil compatible con distintos dispositivos. También se requieren conocimientos en la tecnología Java JEE que permitirán desarrollar la lógica del negocio del lado servidor.

- **Justificación metodológica:**

Con el uso de RUP y el lenguaje de modelado unificado se pretende construir un software robusto orientado a objetos utilizando buenas

prácticas en la fase de ingeniería de software, asegurando la calidad del producto durante las fases de análisis, diseño y desarrollo.

- **Justificación económica:**

El uso de la aplicación no generará costos a los usuarios, la instalación de la aplicación móvil se realizará transfiriendo un archivo al dispositivo y en su ejecución utilizará la red inalámbrica existente en el campus. La universidad no tendrá que realizar inversiones para el funcionamiento de este sistema.

- **Justificación social:**

Éste proyecto será de gran utilidad para la comunidad académica porque complementará el sistema existente de la biblioteca con una aplicación que utiliza tecnologías inalámbricas disponibles en los dispositivos móviles actuales.

- **Justificación profesional:**

Se aplicarán los conceptos generales de JME y RUP adquiridos durante la carrera y los conceptos de JEE como EJB (Enterprise Java Bean), servicios web y JPA (Api de Persistencia de Java) en actividades profesionales que serán utilizados para desarrollar una aplicación innovadora.

- **Justificación investigativa:**

Para el desarrollo del proyecto es necesario consultar bibliografías para profundizar el concepto de metodologías relacionadas con el desarrollo de aplicaciones móviles que consumen un servicio web, ya que este es parte fundamental en la comunicación entre el dispositivo móvil y el servidor de aplicaciones.

- **Justificación legal:**

Se utilizará software libre como alternativa para no incurrir en el uso de software propietario ni en compra de licencias. De esta forma se asegura que los usuarios no tengan que pagar ningún valor por el uso del software resultante de este proyecto.

1.3 OBJETIVOS

1.3.1 General

Desarrollo de una aplicación móvil que permita realizar consultas bibliográficas para la biblioteca de la Universidad Libre sede Bosque Popular.

1.3.2 Específicos

- Evaluar el Sistema de Consultas actual de la Biblioteca de la Universidad.
- Aplicar una encuesta para determinar el nivel de satisfacción de los usuarios frente al sistema actual.
- Determinar los modelos de teléfonos de los diferentes fabricantes que son compatibles con el perfil MIDP 2.0 y CLDC 1.1.
- Desarrollar una interfaz gráfica de fácil manejo y navegabilidad.
- Implementar la lógica de negocio haciendo uso de EJB especificación 3, que hace parte de la tecnología Java EE.
- Implementar las consultas de bibliografía utilizando el API de persistencia de Java (JPA).
- Implementar el servicio web utilizando el API para servicios web XML de Java JAX-WS.

- Implementar un cliente que consuma servicio web que facilite la comunicación entre el dispositivo móvil y la aplicación del lado servidor.
- Realizar pruebas a la aplicación usando un emulador de dispositivo móvil durante la etapa de desarrollo y pruebas.
- Diseñar un ambiente de prueba real para el prototipo de la aplicación.

1.4 ALCANCE

En este proyecto se desarrollará una aplicación móvil con la tecnología Java JME aplicando la metodología RUP (Proceso Unificado de Rational) que permita realizar consultas de bibliografía por título, autor y tema. Adicionalmente el usuario podrá realizar consultas avanzadas donde se combinan los tipos de consulta mencionados anteriormente. El software se entregará con los manuales que contienen toda la información relacionada con el uso e instalación.

La aplicación podrá ser utilizada por los estudiantes y docentes de la Universidad Libre Sede Bosque Popular ubicado en Bogotá, y se instalará en la memoria de los dispositivos que cuenten con la capacidad de conexión a redes móviles o inalámbricas ya que se requiere acceso a Internet para su funcionamiento.

Con su implementación se pretende complementar el sistema de consultas que se usa actualmente en la biblioteca y solo se requerirá el montaje de un servidor de aplicaciones donde se alojará el servicio web y el motor de base de datos donde se mantendrá la información.

1.5 HIPÓTESIS

Ante la disminución en el uso del Sistema de Información de Consultas de la Biblioteca por causas tales como la falta de tiempo por parte de los usuarios,

el desconocimiento del Sistema, falta de recursos suficientes para acceder a la información, una base de datos desactualizada y la complejidad al realizar búsquedas, se presenta la oportunidad de enfrentar los problemas mencionados con una aplicación que permite a los usuarios acceder de una manera más simple a la información debidamente actualizada desde la comodidad de un dispositivo móvil en el momento y lugar deseado.

1.6 VARIABLES

La tabla No. 1 contiene las variables que hacen parte del tema de investigación.

Tabla 1 - Variables

<ul style="list-style-type: none"> • Aplicación Móvil • Usuarios • Conocimiento del sistema actual • Facilidad de uso • Disponibilidad de recursos • Falta de tiempo • Actualización de información 		
Variable Independiente	Variable Dependiente	Variable Interviniente.
Conocimiento del Sistema actual	Aplicación Móvil para consulta de bibliografía	Usuarios
Facilidad de uso		Sistema actual de consultas de la Biblioteca
Disponibilidad de recursos		
Falta de tiempo		
Actualización de información		

Fuente: Elaboración propia

1.7 DISEÑO METODOLÓGICO

En este aparte se describirán el tipo y método de investigación seleccionados para el desarrollo del proyecto.

1.7.1 Tipo de Investigación

Se selecciona el tipo de investigación cuantitativo porque se buscan establecer las causas del poco uso de la aplicación web existente por parte de la comunidad académica perteneciente a la Universidad Libre Sede Bosque Popular por medio de una encuesta. La construcción de una aplicación móvil servirá para probar la hipótesis establecida en el proyecto.

1.7.2 Método de Investigación

Con el método experimental se busca determinar el efecto de una o más variables independientes sobre una variable dependiente. Aplicándose al contexto del proyecto, el método experimental ayuda a determinar si la facilidad de uso y la disponibilidad de recursos necesarios son factores determinantes al momento de la utilización de la aplicación por parte de los usuarios.

1.7.3 Universo

Serán los estudiantes y docentes de diferentes facultades ubicados en la Universidad Libre sede Bosque Popular en la ciudad de Bogotá.

1.7.4 Muestra

Será de 45 Estudiantes y 2 Docentes de diferentes facultades pertenecientes a la Universidad Libre Sede Bosque Popular. Por medio de ellos se obtendrá información sobre el conocimiento y el uso que se le da al sistema actual de consultas de bibliografía.

1.7.5 Fuentes de Información

La encuesta y la entrevista fueron utilizadas como fuentes de información primaria que permiten la recolección de información necesaria para el proyecto.

1.7.6 Instrumentos y Análisis

La entrevista fue realizada a Liliana López, directora de la Biblioteca de la Universidad Libre Sede Bosque Popular, la cual se llevó a cabo en la oficina de la dirección ubicada dentro de las instalaciones de la biblioteca. Por medio de esta se obtuvo información sobre las estadísticas de consultas realizadas por los usuarios en el sistema actual y en las bases de datos externas junto a la descripción de los procesos internos de la biblioteca:

- El plan de compra de libros.
- El proceso de préstamo de libros en bibliotecas públicas y de otras universidades
- La relación de las búsquedas de bibliografía que utiliza el sistema actual.

La entrevista y su contenido se pueden consultar en el anexo A.

La encuesta se aplicó a estudiantes y docentes de diferentes facultades ubicadas en la Universidad Libre sede Bosque Popular. Está compuesto por preguntas cerradas orientadas a obtener información sobre el conocimiento que poseen sobre el sistema de consultas de bibliografía que se utiliza actualmente.

El formato utilizado para el desarrollo de la encuesta se puede consultar en el anexo B.

Al preguntar acerca del conocimiento del sistema actual de consulta de bibliografía, el 57.44% de los encuestados negaron conocerlo frente a un 42.55% que sí lo conoce y de éstos un 38.29% realiza consultas frecuentes a través del sistema. Un 57.44% de los usuarios considera que el sistema actual de consulta es apropiado y a través de éste un 36.17% accede a los catálogos nacionales e internacionales. Finalmente un 93.61% cree que la implementación de una aplicación móvil para la consulta de bibliografía sería una herramienta útil.

Los resultados obtenidos de la encuesta se pueden consultar en el anexo C.

2. MARCO REFERENCIAL

El capítulo que se describe a continuación está compuesto por el marco histórico, teórico, conceptual, legal y tecnológico. El marco histórico relaciona los proyectos desarrollados a nivel nacional e internacional que fueron tomados en cuenta dada la importancia de los conceptos y tecnologías usados en cada uno. Dentro del marco teórico se agrupan los conceptos y teorías utilizados para el desarrollo del argumento del presente proyecto, así mismo, el marco conceptual recoge los conceptos propios sustentados en bases de datos especializadas y su interpretación utilizados dentro del tema de investigación. El marco legal está compuesto por normas que protegen y limitan el uso de tecnologías existentes y la propiedad intelectual. El marco tecnológico describe las distintas herramientas de software que facilitan el desarrollo del mismo. A continuación se referencia cada uno de los marcos mencionados anteriormente.

2.1 MARCO HISTÓRICO

En el desarrollo del marco histórico es necesario mencionar y describir de forma general los antecedentes realizados tanto a nivel local como internacional, incluyendo las tecnologías utilizadas y los resultados obtenidos en cada uno de ellos.

Luego de consultar distintas fuentes, encontramos que se ha desarrollado un proyecto a nivel local que se describe a continuación:

- Grupo de Investigación en Telecomunicaciones GITUC de la Universidad Central, Pedro Luis Ramírez
 - En el año 2006 desarrolló el proyecto llamado "Consulta de libros, reservas y préstamos de libros a una base de datos

de la biblioteca mediante conexión GPRS y Bluetooth" (RAMÍREZ, 2006).

Del proyecto en mención se retoma el uso de las redes móviles (HSDPA, GPRS, EDGE) o inalámbricas (Wi-Fi) para la conexión a internet de la aplicación móvil que permite la consulta de bibliografía. Actualmente, distintos dispositivos que se encuentran en el mercado ya cuentan con la capacidad de conexión y lo ideal es que el usuario no incurra en gastos adicionales mientras hace uso de la aplicación.

En otra de las fuentes consultadas se encuentra que a nivel internacional también se han desarrollado proyectos que involucran el uso de aplicaciones móviles para la consulta de bibliografía, el cual se describe a continuación:

- “El Midlet de la Biblioteca de Ingeniería” (VERA, J. e YBARRA, G. B., 2011)
 - La Facultad de Ingeniería de la Universidad Nacional de la Plata (Argentina) desarrolló la aplicación móvil “BibliIng” usando la tecnología Java JME que brinda las siguientes funciones:
 - Consulta de fecha de vencimiento de préstamos.
 - Consulta la cola de reservas.
 - Posibilidad de renovar los préstamos que estén por vencer.
 - La aplicación puede ser copiada directamente en el dispositivo a través de un cable de datos o bluetooth. También puede ser descargada ingresando a través del navegador wap del dispositivo a la dirección <http://biblio.ing.unlp.edu.ar/JME/> y seleccionando la opción “Instalar BibliIng”.

- Los requisitos mínimos para que “Bibliolng” funcione correctamente son los siguientes:
 - Configuración CLDC (Connected Limited Device Configuration) 1.0.
 - Perfil MIDP (Mobile Information Device Profile) 2.0.
- “Bibliolng” realiza la consulta de préstamos y reservas accediendo a Internet vía celular, esto quiere decir que el usuario debe contar con saldo suficiente en la cuenta para realizar las diferentes operaciones. Actualmente se encuentra en la Versión 0.2

Del proyecto en mención se toma la idea de implementar la configuración CLDC 1.0 y del perfil MIDP 2.0 en el desarrollo de la aplicación móvil para la consulta de bibliografía, de tal forma que se garantice la compatibilidad en diferentes dispositivos móviles.

2.2 MARCO TEÓRICO

En el desarrollo del marco teórico se aplican distintas teorías que desde el punto de vista profesional permiten la elaboración del contenido del proyecto, las cuales serán descritas a continuación:

- El desarrollo de software orientado a objetos permite aplicar los conceptos de herencia, polimorfismo y encapsulación a los objetos que son una representación de la realidad. Por ejemplo, un libro que se encuentra en los estantes de la biblioteca cuenta con ISBN, título, autor, tema, edición, editorial, número de ejemplar los cuales serán atributos del objeto dentro del software junto a los métodos u operaciones “consultar” y “reservar”.
- Se implementará una arquitectura de tres capas conocida como modelo vista controlador (MVC) que es un patrón de diseño que separa la interface del usuario, la lógica de control y los datos en

distintos componentes. Aplicado en el proyecto, la interface de usuario y la lógica del control estarán en el dispositivo móvil mientras que los datos estarán del lado servidor.

La lógica de control es la encargada de enviar las peticiones ingresadas por el usuario en la vista al servicio web, el cual se comunica con los componentes EJB que hacen parte del modelo y que procesa las peticiones. El modelo genera una respuesta que retorna al control a través del servicio web donde será procesada y mostrada en la vista al usuario.

- Se hará uso del Proceso Unificado de Rational (RUP), el cual se fundamenta en el uso de las mejores prácticas como el desarrollo iterativo, el seguimiento a los requerimientos por medio de UML (Lenguaje Unificado de Modelado), el uso de arquitecturas que permiten la reutilización de código y la continua verificación de calidad del producto. Esta metodología está compuesta por cuatro fases que incluyen las actividades relacionadas con la ingeniería de software:
 - En el desarrollo de la fase de inicio, las personas interesadas buscan establecer el enfoque del sistema que se va a desarrollar, se determinan los casos de uso críticos o funcionalidades más importantes, se propone una arquitectura candidata que soportará el funcionamiento del sistema y los riesgos potenciales que se puedan presentar.
 - Durante la fase de elaboración se debe asegurar que los riesgos relacionados con la arquitectura estén resueltos para luego implementar los requerimientos en los prototipos que serán utilizados de manera exploratoria, para mitigar riesgos o para realizar demostraciones.
 - En la fase de construcción se realizan las iteraciones necesarias para completar el análisis, diseño, desarrollo y

pruebas de las funcionalidades que componen el sistema hasta obtener la versión que será entregada a los usuarios.

- Finalmente, en la fase de transición el usuario realizará pruebas de usabilidad y rendimiento sobre el nuevo sistema que le permitirán validar las expectativas. Los problemas que sean encontrados luego de las pruebas serán corregidos en la versión que entrará a producción. Adicionalmente serán entregados los manuales técnicos y de uso.

2.3 MARCO CONCEPTUAL

Los conceptos técnicos y científicos que se aplicarán a lo largo del proyecto serán definidos de acuerdo a una entidad reconocida, que en este caso será la IEEE (Instituto de Ingenieros Eléctricos y Electrónicos).

La arquitectura de software es la estructura de los componentes de un programa o sistema, sus interrelaciones, principios y directrices que rigen su diseño y evolución en el tiempo.

La compatibilidad es la habilidad de dos o más sistemas para realizar sus funciones mientras comparten el mismo hardware o software.

Una base de datos es una colección de datos interrelacionados almacenados en archivos computarizados.

Un diseño orientado a objetos es una técnica de desarrollo de software donde un sistema o componente es expresado en término de objetos y conexiones entre ellos.

El lenguaje orientado a objetos permite al usuario expresar un programa en términos de objetos y mensajes entre ellos.

Un protocolo es un conjunto de convenciones que dirigen la interacción de procesos, dispositivos y otros componentes dentro de un sistema.

Un prototipo es un tipo, forma o instancia de un sistema preliminar que sirve como un modelo para un estado posterior o final del sistema.

2.4 MARCO LEGAL

- Las aplicaciones desarrolladas en este proyecto no generarán interferencia a otras aplicaciones instaladas.
- La aplicación móvil hará uso de redes móviles o inalámbricas para la transferencia de información relacionada con las consultas de bibliografía. En ningún momento se tendrá acceso a otra información almacenada en el dispositivo.
- La conectividad y uso de redes inalámbricas por parte de los dispositivos móviles que cuenten con esta característica estarán bajo la normatividad del estándar 802.11 b y g de la IEEE.
- No habrá reproducción total o parcial de código fuente perteneciente a terceros ya que todo será producido por el involucrado en el proyecto.
- La propiedad intelectual de este proyecto estará sujeta a las normas vigentes establecidas por la Universidad Libre.

2.5 MARCO TECNOLÓGICO

A continuación se describirán las tecnologías utilizadas, la herramienta para el diseño de aplicación, el entorno de desarrollo y el motor de base de datos utilizados en la elaboración del proyecto.

- La tecnología Java JME (Java 2 Micro Edition), la cual permite el desarrollo de aplicaciones móviles compatibles con el software y hardware de diversos fabricantes. Se utilizará esta tecnología para

construir una aplicación móvil compatible con distintos dispositivos que tengan la capacidad de conexión a redes inalámbricas basadas en el estándar 802.11 b y g.

Ilustración 1 - Estructura de la tecnología JME



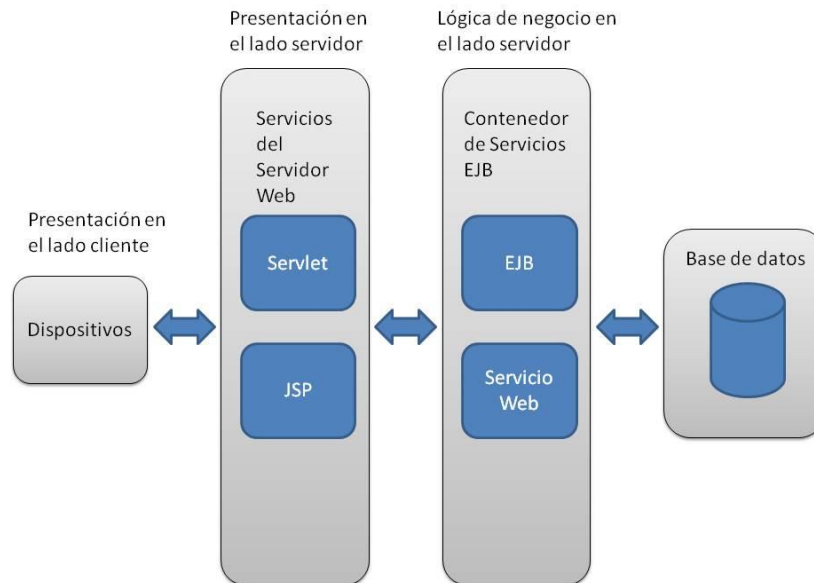
Fuente: Elaboración propia

La ilustración No. 1 representa la estructura de la tecnología JME existente en un dispositivo móvil. La máquina virtual KVM se encuentra en el mismo nivel del sistema operativo mientras que la configuración CLDC y el perfil MIDP hacen parte de las librerías pertenecientes a la tecnología. Por último encontramos los MIDlets que son las clases responsables del ciclo de vida de una aplicación móvil.

- La tecnología Java JEE (Java Enterprise Edition), que permite la construcción y despliegue de aplicaciones empresariales robustas, escalables y mantenibles. Contiene un conjunto de APIs de las

cuales se utilizarán EJB, JPA y servicios web para implementar la lógica del negocio de la aplicación en el lado servidor.

Ilustración 2 - Especificación de la tecnología JEE



Fuente: Elaboración propia

La ilustración No. 2 representa las especificaciones pertenecientes a la tecnología JEE en el lado servidor y cliente. La lógica del negocio del lado del servidor se puede implementar gracias a los servicios web y EJB mientras que la presentación en el lado del servidor se puede implementar mediante el uso de Servlets o JSP.

- Se hace uso del programa Visual Paradigm for UML versión 8.1 en su versión libre para la elaboración de diferentes diagramas UML.

Ilustración 3 - Logo de Visual Paradigm



Fuente: Tomada del sitio <http://www.visual-paradigm.com/>

- Se utiliza la versión 6.7 del ambiente de desarrollo integrado NetBeans, el cual facilita la construcción y pruebas de aplicaciones Java integrando diferentes tecnologías. En este caso permite integrar una aplicación de tipo empresarial, compuesta por EJB, con una aplicación móvil por medio de un servicio web.

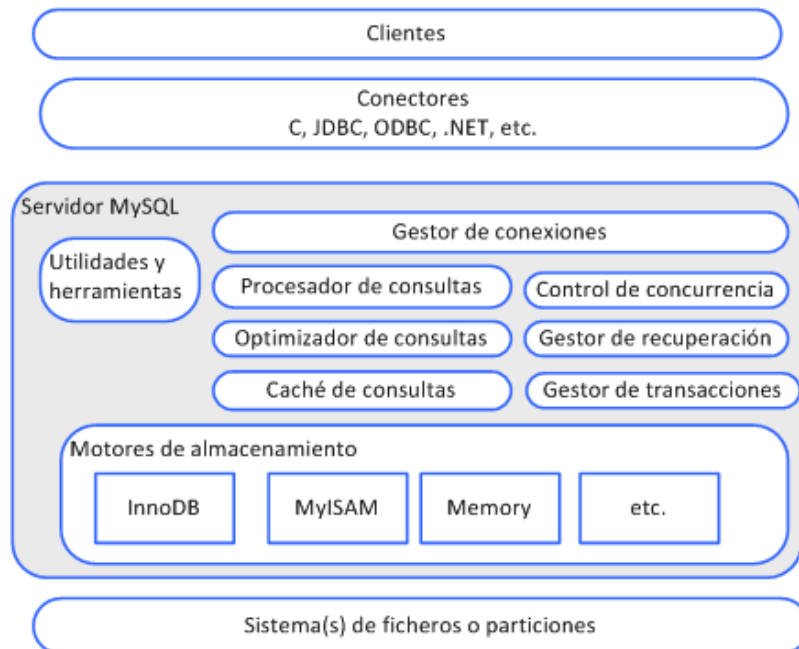
Ilustración 4 - Logo del ambiente de desarrollo integrado Netbeans



Fuente: Tomada de <http://netbeans.org/index.html>

- Para el almacenamiento de datos se utilizará la versión comunitaria 5.1 del motor de base de datos MySQL, el cual pertenece a un proyecto de código fuente abierto al público administrado por Oracle.

Ilustración 5 - Arquitectura del motor de base de datos MySQL



Fuente: Tomada de <http://cnx.org/content/m18938/latest>

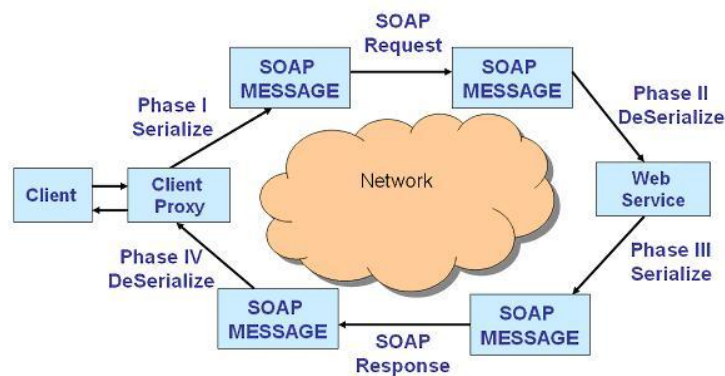
La ilustración No. 5, representa la arquitectura del motor de base de datos que está compuesta por:

- El gestor de conexiones que permite mantener o eliminar la conexión del cliente.
- El caché de consultas se encarga de guardar las consultas y sus resultados.
- El control de concurrencias evita lectura o escritura simultánea de los mismos datos para evitar inconsistencias.
- La gestión de transacción dota de semántica a una consulta que se declara como una sola transacción.
- La gestión de recuperación permite volver hacia atrás las partes de una transacción.
- Los motores de almacenamiento permiten el buen funcionamiento tanto del manejador de disco como de archivo a través de diferentes tablas y registros.

- El intercambio de información entre la aplicación móvil y la lógica del negocio será posible gracias al protocolo SOAP (Protocolo de Acceso a Objetos Simples), el cual está compuesto por mensajes XML que contienen la información de la consulta enviada por la aplicación móvil y la respuesta de las consultas.

Ilustración 6 - Arquitectura de un servicio web

XML Web Services Architecture



Fuente: Tomada de <http://paradithanq.blogspot.com/2011/03/module-2-arsitektur-xml-web-service.html>

La ilustración No. 6 representa el funcionamiento de un servicio web. Cuando el cliente envía una solicitud, ésta se envuelve en un mensaje SOAP (Simple Object Access Protocol) y debe ser serializado para viajar en la red. Cuando el mensaje llega al servicio web debe ser deserializado para poder ser interpretado. La respuesta del servicio web también debe envolverse en un mensaje SOAP y ser serializado para el viaje de retorno al cliente donde será deserializado e interpretado.

3. ESTRUCTURA TEMÁTICA

3.1 INGENIERÍA DEL PROYECTO

Para el desarrollo de este proyecto se aplicará la metodología RUP (Proceso Unificado de Rational), la cual fue descrita en el capítulo 3 sección 3.2. Ésta metodología se fundamenta en el uso de las mejores prácticas como el desarrollo iterativo, el seguimiento a los requerimientos por medio de UML (Lenguaje de Modelado Unificado) además de destacar el uso de arquitecturas que permiten la reutilización de código y la continua verificación de calidad del producto.

A continuación se describen las actividades realizadas en cada una de las fases que hacen parte de la metodología RUP.

3.1.1 Fase de Inicio

En el desarrollo de fase de inicio se deben tener en cuenta los riesgos asociados con el estado del sistema actual, debido a la necesidad de conocer y entender su estructura y funcionamiento. En consecuencia, se realiza una breve descripción del sistema que incluye su interacción con los usuarios y revela la arquitectura sobre la cual está construida la aplicación. Además se relaciona el diagrama de modelo de negocio que está compuesto por las entradas que corresponden a los datos ingresados por el usuario para realizar las diferentes consultas, los procesos de consulta de base de datos y las salidas compuestas por tablas que incluyen los resultados de las consultas, todo lo anterior está involucrado en el funcionamiento del sistema. Adicionalmente, se usan los diagramas y plantillas de casos de uso que servirán para la descripción detallada del proceso de consulta y solicitud de préstamo de bibliografía que hacen parte

de los requerimientos. Para culminar la descripción de esta fase, se continúa con la fase de elaboración donde se describirán los casos de uso y las especificaciones que hacen parte del sistema propuesto.

A continuación se mencionan los componentes descritos anteriormente.

Riesgos detectados en la fase de inicio

La tabla No. 2 contiene los riesgos detectados durante el desarrollo de la fase de inicio, que son clasificados según su magnitud y cada uno debe estar acompañado de un plan de contingencia.

Tabla 2 - Riesgos en la fase de inicio

Número de riesgo	Clasificación del Riesgo / Magnitud	Descripción del Riesgo e Impacto	Estrategia de Mitigación / Plan de Contingencia
RI01	4	Dificultad para realizar la descripción del sistema actual y la formulación de los casos de uso correspondientes	Acceder a cada una de las funcionalidades que ofrece el sistema para conocer con más detalle su funcionamiento.
RI02	4	Se van a definir las especificaciones de un sistema que hará uso de nuevas tecnologías.	Es necesario estudiar las posibles tecnologías que se implementarán en el sistema y de esta forma se pueden establecer las especificaciones.

Fuente: Elaboración propia

Para mitigar el riesgo RI01 asociado con la descripción del sistema actual y la formulación de los casos de uso se decide ingresar al catálogo en línea ubicado en la dirección <http://consulbiblo.unilibre.edu.co:81/reservacion/index.php>, donde se navegará por las funcionalidades de búsqueda simple y avanzada siguiendo

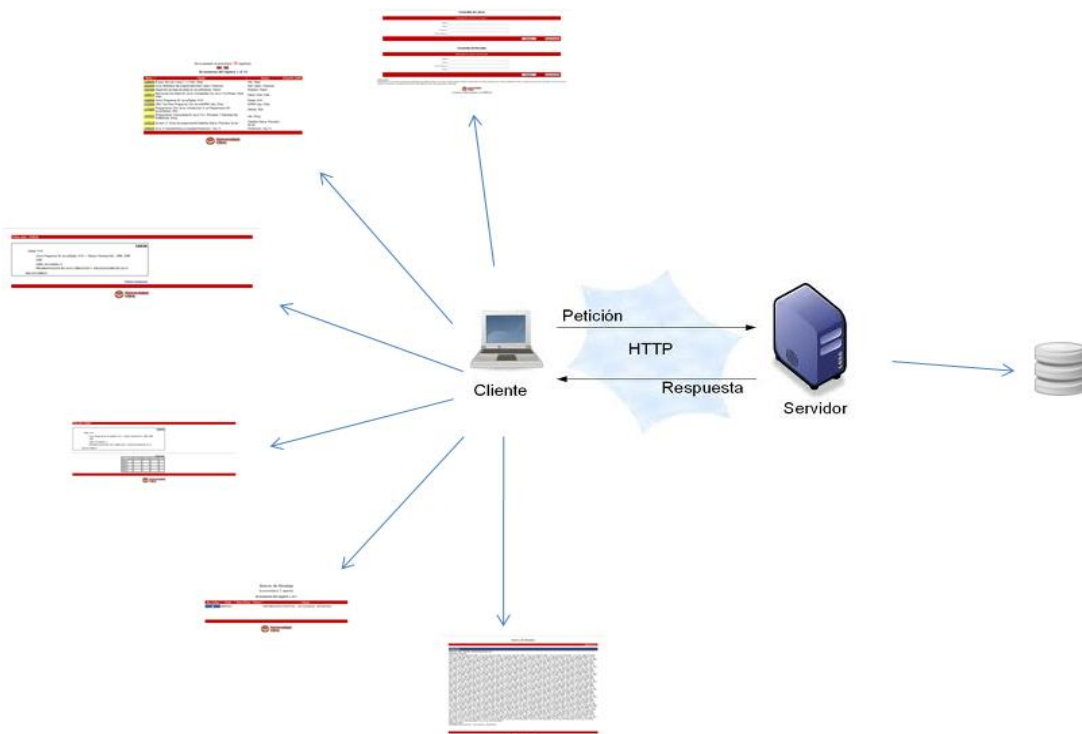
las instrucciones proporcionadas con el fin de adquirir el conocimiento base sobre su funcionamiento y los servicios que ofrece a los usuarios. El resultado de esta actividad se puede evidenciar en el desarrollo de la fase de inicio.

Descripción del Sistema Actual

- Los estudiantes y docentes de la Universidad Libre Sede Bosque Popular tienen acceso al sistema de consultas a través de los equipos de cómputo ubicados en las instalaciones de la Universidad, que en el contexto del proyecto serán denominados “el cliente”. En el cliente se visualizan los formularios donde los usuarios podrán ingresar los datos del libro o revista a consultar y luego de pulsar el botón de búsqueda será enviada una petición al servidor para realizar la consulta.
- El servidor recibe la petición y de ésta se extrae la información ingresada en los formularios para ser pasada a la base de datos donde se realizará la consulta. El resultado será incluido en la respuesta que será enviada de regreso al cliente.
- El cliente recibe la respuesta y luego es interpretada de tal forma que muestra el resultado de la consulta en una página indicando el número de registros y los libros o revistas que coinciden con el(los) parámetro(s) ingresado(s).
- Si el usuario desea conocer la ficha bibliográfica y las existencias del libro o revista, pulsando el enlace ubicado junto al resultado, se envía otra petición de consulta al servidor con el número del material.
- La respuesta del servidor es interpretada nuevamente en el cliente y muestra una página con la información relacionada al material seleccionado.

La ilustración No. 7 muestra la implementación de la arquitectura cliente-servidor en el sistema actual de consultas de bibliografía.

Ilustración 7 - Arquitectura cliente-servidor del sistema actual

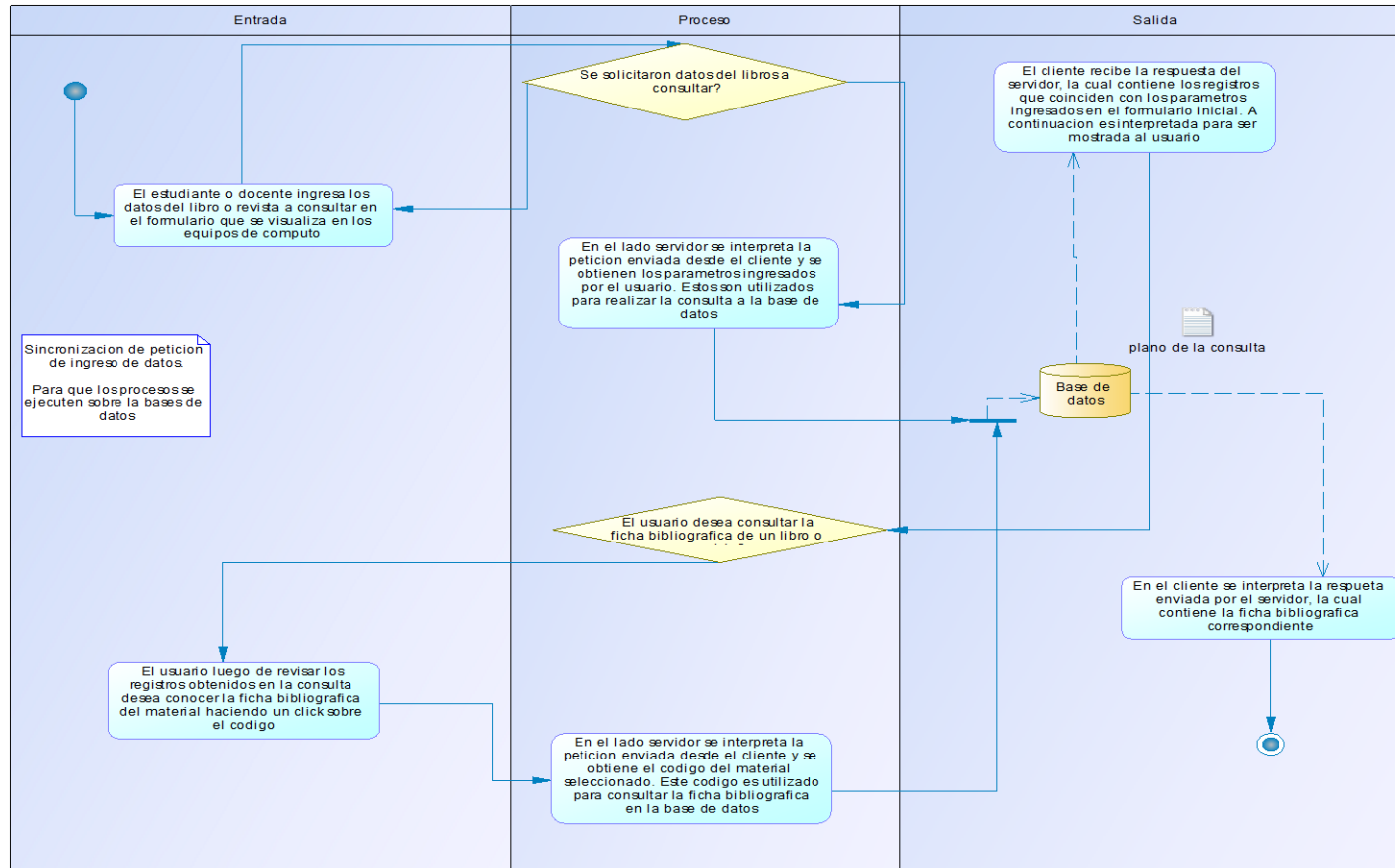


Fuente: Elaboración propia

Modelo de negocio

La ilustración No. 8 representa el modelo de negocio del sistema actual que está compuesto por:

Ilustración 8 - Modelo de negocio del sistema actual



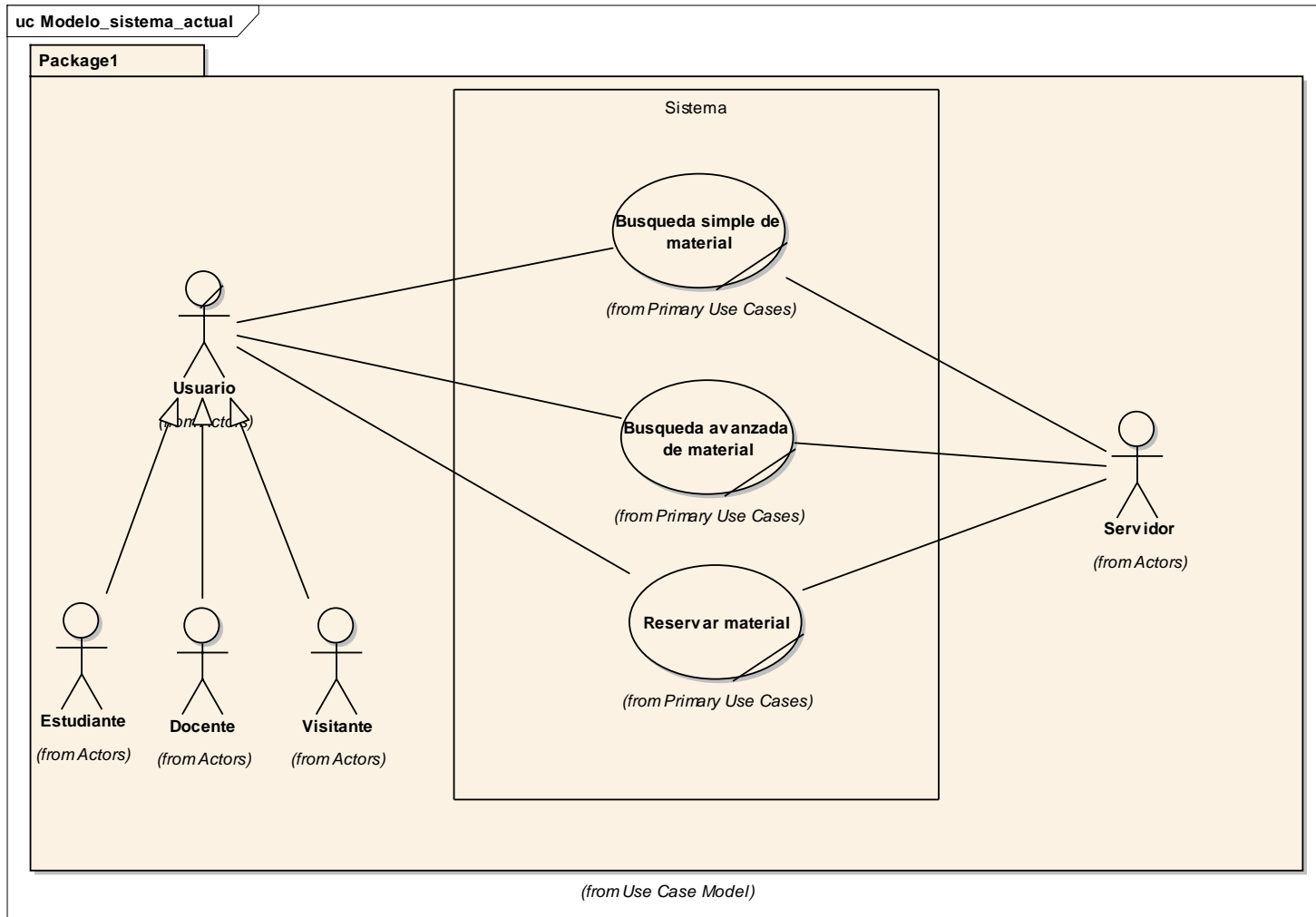
Fuente: Elaboración propia

Casos de uso de consulta y reserva

Desde el punto de vista de la ingeniería de software, los casos de uso describen paso a paso y de forma detallada la interacción entre los usuarios (actores) y el sistema. Por otra parte, los diagramas de caso de uso representan de forma gráfica el comportamiento de un sistema desde la perspectiva del usuario y debido a esto su interpretación es más sencilla.

La ilustración No. 9 representa el diagrama de casos de uso que conforman el sistema actual:

Ilustración 9 - Diagrama de casos de uso del sistema actual



Fuente: Elaboración propia

Para describir los casos de uso se emplea un formato compuesto por:

- Los actores que participan
- Una descripción general
- Las precondiciones necesarias para la ejecución
- La descripción del flujo principal de eventos
- Flujos alternativos al principal
- Las post condiciones resultantes a la ejecución
- Flujos excepcionales
- La frecuencia de uso y su importancia

A continuación se encuentran las plantillas de casos de uso, en las cuales se realiza una descripción del caso de uso donde se incluyen los actores, la descripción del caso de uso, las precondiciones, el flujo principal acompañado de las acciones del actor y del sistema, las post condiciones, el flujo excepcional al flujo principal, la frecuencia y su importancia.

La tabla No. 3 contiene la plantilla para el caso de uso de consulta simple del material. Este caso de uso consiste en la interacción del usuario con el sistema actual para realizar consultas simples de material.

Tabla 3 - Caso de uso para búsqueda simple de material

Nombre del Caso de Uso	Búsqueda simple de material	
Código del Caso de Uso	1	
Actor(es)	Usuario, Servidor	
Descripción	Este caso describe las acciones correspondientes a la consulta de bibliografía en el sistema actual. El caso de uso inicia cuando el usuario ingresa en el campo del formulario la expresión del material a buscar y termina cuando el servidor procesa la respuesta para ser mostrada al usuario	
Precondición	La necesidad de los usuarios para conocer las existencias del material bibliográfico que posee la biblioteca de la Universidad Libre sede Bosque Popular	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario digita la dirección del sistema de consultas en la barra de direcciones del navegador web o accede a través de los equipos ubicados en la biblioteca	2. Muestra el formulario de consulta general.
	3. El usuario digita la expresión relacionada con el material a	4. El sistema realiza la consulta basada en el parámetro

	buscar en el campo y da un click sobre el botón "Buscar"	ingresado y muestra el resultado al usuario en una tabla con las siguientes columnas: Clasificación, Título, Autor
	5. El usuario hace click sobre el registro para mostrar la ficha del material	6. El sistema muestra un recuadro con diferentes tipos de visualización: Ficha catalográfica, Tabla, Marc y Personalizada. Si el material seleccionado es un libro, se muestra la siguiente información: Autor, Título Libro, Autor, Editorial, Edición, Número de páginas, ISBN, Temática, No. Topográfico. En la parte inferior se encuentra una tabla con las existencias del material: Estado, Biblioteca, No. Adqui, Volumen, Ejemplar, Tomo. Si el material seleccionado es una revista, se muestra la siguiente información: Título, Editorial, Temática. En la parte inferior se encuentra una tabla con las existencias del material: Estado, Biblioteca, No. Adqui, Volumen, Número, Año
	7. El usuario puede realizar la reserva del material que está consultando haciendo un click sobre el enlace "Reservar"	8. El sistema muestra un formulario donde deberá ser ingresado el No. De cuenta y el NIP del usuario para

	ubicado en la tabla de existencias.	realizar la reserva.
	9. El usuario ingresa el No. De cuenta y el NIP en los campos del formulario y hace un click en el botón "Reservar" para continuar con el proceso.	10.El sistema realiza la reserva del material para el usuario
	11.El usuario deberá acercarse a las instalaciones de la biblioteca para hacer efectivo el préstamo.	
Post condición	La aplicación muestra un enlace en la parte superior del formulario que llevará al usuario nuevamente al formulario de consulta simple.	
Flujo Excepcional	1. El usuario ingresa la expresión del material a consultar	2. El sistema muestra un mensaje indicando al usuario que no se encontró ningún resultado con el parámetro de búsqueda proporcionado
	3. El usuario ingresa la expresión del material a consultar	4. El sistema muestra un mensaje indicando que se ha presentado un error de comunicación
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 4 contiene la plantilla para el caso de uso de consulta avanzada de material. Este caso de uso consiste en la interacción del usuario con el sistema para realizar consultas avanzadas de material

Tabla 4 - Caso de uso para la consulta avanzada de material

Nombre del Caso de Uso	Búsqueda avanzada de material	
Código del Caso de Uso	2	
Actor(es)	Usuario, Servidor	
Descripción	Este caso describe las acciones correspondientes a la consulta avanzada de material en el sistema actual. El caso de uso inicia cuando el usuario ingresa las expresiones a buscar seleccionando de las listas los campos donde pertenecen dichas expresiones, la conjunción o disyunción correspondiente, la ubicación donde se realizará la búsqueda y el tipo de material. El caso de uso termina cuando el servidor procesa la respuesta para ser mostrada al usuario.	
Precondición	La búsqueda simple no arroja los resultados esperados y el usuario posee información específica sobre el material a consultar	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario digita la dirección del sistema de consultas en la barra de direcciones del navegador web o accede a través de los equipos ubicados en la biblioteca	2. Muestra el formulario de consultas
	3. El usuario selecciona la opción "Búsqueda Avanzada" ubicada en la parte superior	4. Se muestra el formulario de consulta avanzada

	del formulario	
	<p>5. El usuario puede ingresar la expresión a localizar en cada caja de texto del formulario y puede seleccionar de las listas el campo donde quiere realizar la consulta (en cualquier campo, en títulos, en autores, en temas, en ISBN, en Editoriales, en Series, en Clasificación, en Notas, Fecha de publicación, Fecha de copyright, Lugar de publicación) para que sea más específica. Es posible seleccionar de las listas el operador booleano (And, Or, Not) para la búsqueda de prefijos o raíces. La búsqueda se puede realizar en distintas bases de datos, las cuales se pueden seleccionar de una lista. El tipo de material a consultar también se puede seleccionar de una lista (Cualquier material, Libro, Tesis, Mapas, Video, CD Rom/DVD, Fotografías, Diapositivas, Disquetes, Música, Microfilm, Archivo de PC, Pintura, Escultura,</p>	<p>6. El sistema realiza la consulta basada en los parámetros ingresados y muestra el resultado al usuario en una tabla con las siguientes columnas: Clasificación, Título y Autor</p>

	Revistas, Otros). Para iniciar la consulta da un click sobre el botón "Buscar".	
	7. El usuario puede dar un click sobre los registros resultantes para conocer la ficha bibliográfica.	8. La ficha puede ser visualizada en diferentes tipos: Ficha catalográfica, Tabla, MARC, Personalizada. Se observa la siguiente información relacionada con el material seleccionado: Autor, Título, Editorial, Edición, ISBN, Temática. En la parte inferior aparece una tabla con las existencias del material seleccionado: Estado, Biblioteca, No. Adqui, Volumen, Ejemplar, Tomo
	9. El usuario puede hacer la reserva del material consultado haciendo un click sobre el enlace "Reservar" ubicado en la tabla de existencias.	10. El sistema muestra un formulario donde deberá ser ingresado el No. de cuenta y el NIP del usuario para realizar la reserva.
	12. El usuario ingresa el No. De cuenta y el NIP en los campos del formulario y hace un click en el botón "Reservar" para continuar con el proceso.	13. El sistema realiza la reserva del material para el usuario
	14. El usuario deberá acercarse a las instalaciones de la biblioteca para hacer efectivo el préstamo.	

Post condición	La aplicación muestra un enlace en la parte superior del formulario que llevará al usuario nuevamente al formulario de consulta simple.	
Flujo Excepcional	1. El usuario realiza alguna operación de consulta	2. El sistema muestra un mensaje indicando al usuario que la consulta no arroja resultado alguno
	5. El usuario realiza alguna operación de consulta	6. El sistema muestra un mensaje indicando que se ha presentado un error de comunicación
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 5 contiene la plantilla para el caso de uso de reserva del material consultado. Este caso de uso consiste en la solicitud de reserva del material consultado por el usuario.

Tabla 5 - Caso de uso para la reserva de material

Nombre del Caso de Uso	Reservar material	
Código del Caso de Uso	3	
Actor(es)	Usuario, Servidor	
Descripción	Este caso describe las acciones correspondientes a la reserva de material en el sistema actual. El caso de uso inicia cuando el usuario revisa el resultado obtenido en la búsqueda simple o en la búsqueda avanzada y selecciona el material para ver la ficha bibliográfica para ver las existencias. El caso de uso termina cuando el sistema procesa la solicitud de reserva del usuario.	
Precondición	El usuario debe haber realizado una búsqueda simple o avanzada de material y luego consultar la ficha bibliográfica del material seleccionado.	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario revisa el resultado de la consulta y da un click sobre el registro a revisar	2. Recibe el código del registro seleccionado y muestra la ficha bibliográfica
	3. La ficha bibliográfica contiene la información del material seleccionado y en la parte inferior aparece una tabla con las existencias	4. Recibe la petición de reserva de material y muestra el formulario de ingreso de datos

	disponibles. El usuario hace click sobre el enlace "Reservar"	
	5. El usuario debe ingresar el No. de cuenta y el NIP en el formulario que aparece en pantalla para hacer la reserva del material	6. Recibe los datos ingresados en el formulario para realizar la reserva y se informa al usuario del éxito de la operación
Post condición	La aplicación muestra un enlace en la parte superior del formulario que llevará al usuario nuevamente al formulario de consulta simple.	
Flujo Excepcional	1. El usuario realiza alguna operación de consulta	2. El sistema muestra un mensaje indicando al usuario que la consulta no arroja resultado alguno
	3. El usuario ingresa el No. de cuenta y un NIP	4. El sistema valida que los datos ingresados para la reserva sean correctos. Si los datos no corresponden, se notifica al usuario del error en los datos
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

Requerimientos del Sistema Actual

La tabla No. 6 hace referencia a los requerimientos pertenecientes al sistema actual. Los requerimientos son definidos como la capacidad del software para resolver un problema al que se enfrenta un usuario en el desarrollo de sus actividades y pueden ser de varios tipos:

- Los requerimientos de usuario representan los resultados esperados con el uso del sistema.
- Los requerimientos del sistema incluyen todas las funcionalidades con las que cuenta junto a sus restricciones.

Tabla 6 - Requerimientos del Sistema Actual

Número Requerimiento	Nombre Requerimiento	Descripción
R01	Acceder a la consulta de libros o revistas a través de un navegador de internet.	El acceso al formulario de consultas de libros y revistas se realiza a través de un navegador web.
R02	Desplegar el formulario para la consulta de libros y revistas.	El sistema debe mostrar un formulario con los campos necesarios para la consulta de libros y revistas.
R03	Implementar el uso del protocolo HTTP para la comunicación entre el cliente y el servidor.	Las peticiones enviadas desde el cliente al servidor y sus respuestas son enviadas a través del protocolo HTTP.
R04	Implementar la arquitectura cliente-servidor en la aplicación.	Esta arquitectura requiere que el servidor esté en capacidad de resolver las peticiones realizadas desde el cliente.

R05	El sistema para la consulta de libros y revistas está disponible tanto en la red interna como en la página web de la universidad	Gracias a esto, los usuarios tienen acceso al catálogo de libros y revistas dentro y fuera de las instalaciones de la universidad.
R06	El sistema debe permitir al usuario realizar búsquedas simples.	En este tipo de consulta se deben ingresar la expresión relacionada con el material a consultar en el campo mostrado en el formulario.
R07	El sistema debe permitir al usuario hacer búsquedas avanzadas	En este tipo de consulta se deben mostrar tres campos donde se ingresan las expresiones relacionadas en el material. Adicionalmente se selecciona el campo al que pertenece cada expresión (cualquier campo, título, autor, tema, isbn, editorial, series, clasificación, notas, fecha de publicación, fecha de copyright, lugar de publicación). También es posible hacer uso de operadores booleanos (AND, OR, NOT) para buscar prefijos o raíces.

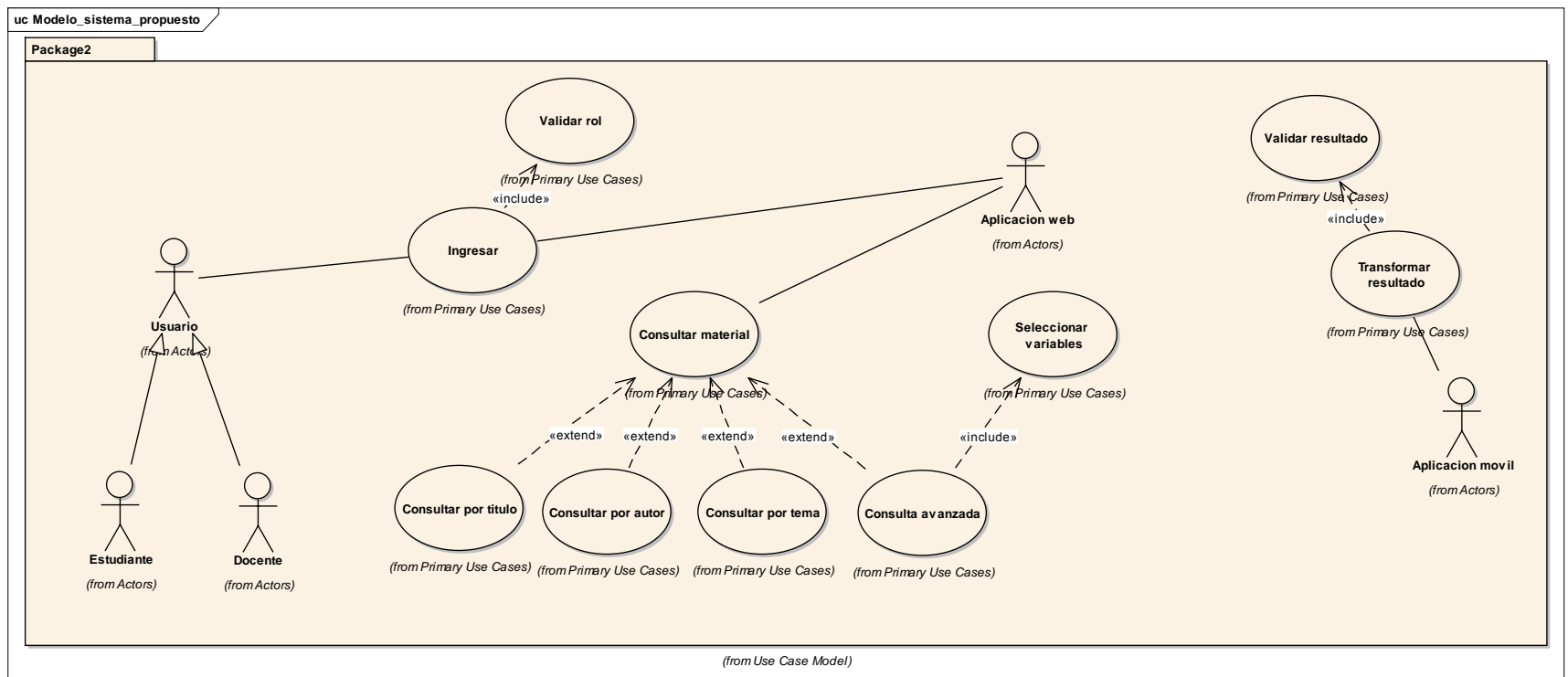
R08	El sistema debe permitir al usuario realizar reserva de material	Luego de realizar la búsqueda del material, el usuario puede solicitar que sea reservado luego de ingresar las credenciales (No. de cuenta y NIP) para ser identificado en el sistema.
-----	------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Elaboración propia

Casos de Uso del Sistema Propuesto

La ilustración No. 10 muestra los casos de uso que hacen parte del sistema propuesto

Ilustración 10 - Diagrama de casos de uso del sistema propuesto



Fuente: Elaboración propia

Especificaciones del Sistema Propuesto

La tabla No. 7 hace referencia a las especificaciones del sistema propuesto que corresponden a los aspectos más relevantes del producto a desarrollar y serán la base para su construcción. Para mitigar el RI02, se estudiaron y seleccionaron diferentes tecnologías para la construcción de la solución: JME para el desarrollo de la aplicación móvil que será usada por la comunidad académica para realizar las consultas bibliográficas, EJB y Servicios Web que hacen parte de JEE para el desarrollo de la lógica de negocio del lado servidor. Las especificaciones reciben un nombre y son acompañadas por una breve descripción.

Tabla 7 - Especificaciones del sistema propuesto

Número Especificación	Nombre Especificación	Descripción
E01	Autenticar el ingreso de usuarios	El usuario debe ingresar nombre de usuario y contraseña para acceder al menú principal de la aplicación.
E02	Identificación del menú principal de la aplicación	Habrà un menú diferente para cada rol que ingrese a la aplicación.
E03	Acceder a las consultas de bibliografía desde la aplicación móvil	Para complementar el sistema actual de consulta de bibliografía y facilitar su uso se desarrolla la aplicación móvil
E04	Desplegar los formularios correspondientes a la opción seleccionada en el menú principal	Los formularios deben contener los campos suficientes para el ingreso de parámetros necesarios para las consultas
E05	La aplicación hará uso de redes móviles	La aplicación móvil se puede instalar en un teléfono que

		tenga capacidades de conexión a redes móviles EDGE, 3G o inalámbricas Wi Fi para realizar la consulta de bibliografía
E06	Compatibilidad de la aplicación móvil con el dispositivo	El dispositivo móvil donde se vaya a instalar la aplicación debe tener unas características mínimas para que haya un funcionamiento óptimo: perfil MIDP 2.0, CLDC 1.1, 128 MB de memoria
E07	Compatibilidad de la aplicación móvil con CLDC 1.1 y MIDP 2.0	La aplicación puede ser instalada en teléfonos móviles que cumplan con esta configuración mínima
E08	Permitir la facilidad de uso de la aplicación móvil por medio de una interfaz gráfica	La interfaz gráfica de la aplicación móvil debe ser amigable para facilitar su uso
E09	Implementar el uso del protocolo HTTP para la comunicación entre el dispositivo móvil y el servidor de aplicaciones	La comunicación entre el dispositivo móvil y el servidor de aplicaciones se realiza por medio del protocolo HTTP
E10	Implementar un servicio web que le permite a la aplicación móvil tener acceso a la lógica del negocio desplegada en la aplicación web	El servicio web expone los métodos que permiten a la aplicación móvil obtener información de bibliografías a partir de los parámetros ingresados por el usuario en la aplicación móvil
E11	Realizar un Intercambio de mensajes por medio del protocolo SOAP entre el servicio web y la aplicación	Este protocolo permite el intercambio de información estructurada entre la aplicación móvil y la aplicación web

	móvil	desplegada en el servidor de aplicaciones
E12	La aplicación móvil permite al usuario realizar consultas por Título, Autor, Tema y Consultas Avanzadas	La aplicación móvil permite a los usuarios realizar consultas de bibliografía por título, autor y tema, además ofrece la funcionalidad de consultas avanzadas, donde se pueden seleccionar 2 ó 3 parámetros de búsqueda
E13	Optimizar el rendimiento de la aplicación móvil para reducir tiempos de respuesta	Se deben tener buenas prácticas de programación para gestionar la memoria del dispositivo y así ofrecer un rendimiento óptimo al usuario
E14	Las aplicaciones móviles y web deben ofrecer a los usuarios estabilidad durante su uso	Las aplicaciones están desarrolladas con la tecnología Java, la cual permite construir un software robusto
E15	Optimizar el rendimiento de la aplicación web para consumir los recursos necesarios en el servidor de aplicaciones	A partir de buenas prácticas de programación se puede lograr un consumo mínimo de los recursos del servidor de aplicaciones como acceso a base de datos y consumo de memoria de la máquina virtual
E16	Optimizar el rendimiento del motor de base de datos para reducir tiempo en las consultas de bibliografía	Esto se logra a través de la normalización de las tablas y orden en las consultas SQL que se realicen
E17	Asegurar una configuración óptima del servidor de aplicaciones para ofrecer un rendimiento aceptable de la	Se configura el pool de conexiones de tal forma que pueda atender a un amplio número de usuarios

	aplicación móvil	
E18	Asegurar una latencia baja durante el uso de la aplicación móvil	Esto se puede lograr al ubicar al servidor de aplicaciones en un canal dedicado a recibir únicamente las peticiones enviadas desde la aplicación móvil y que no se vea afectada por el tráfico en la red de la Universidad
E19	Asegurar que el uso de la aplicación móvil no tendrá ningún costo para los usuarios	Esta es una aplicación hecha para complementar el sistema de consultas existente para beneficiar a la comunidad unilibrista
E20	Ofrecer al usuario información completa luego de realizar las consultas bibliográficas	Los resultados deben ser completos y confiables, de tal forma que genere el hábito de uso de la aplicación

Fuente: Elaboración propia

3.1.2 Fase de Elaboración

En la fase de elaboración se analizan los riesgos que se pueden presentar con la elección de la arquitectura base de la aplicación y posteriormente se ejecuta un plan de contingencia. También se realizan las actividades de análisis y diseño de la solución propuesta, las cuales inician con el modelo de negocio del sistema propuesto que está compuesto por las entradas determinadas por el tipo de consulta seleccionado por el usuario, procesos constituidos por la generación de los formularios y la validación de la información ingresada en ellos y salidas representadas por las consultas a la base de datos involucradas en el funcionamiento del mismo. Los diagramas de casos de uso del sistema propuesto se encargarán de describir la interacción del usuario con la aplicación móvil y la comunicación de ésta con el servicio web que se encuentran alojados en el servidor de aplicaciones instalado en el equipo donde fueron desarrollados los aplicativos. Para concluir con la fase, se incluyen los diagramas de actividad y secuencia que representan el flujo en la aplicación móvil. A continuación se hará la descripción de cada uno de los elementos que componen esta fase.

Riesgos detectados en la fase de elaboración

La tabla No. 8 contiene los riesgos detectados durante el desarrollo de la fase de elaboración que son clasificados según su magnitud y cada uno debe estar acompañado de un plan de contingencia.

Tabla 8 - Riesgos en la fase de elaboración

Número del Riesgo	Clasificación del Riesgo / Magnitud	Descripción del Riesgo/ Impacto	Estrategia de Mitigación y/o Plan de Contingencia
RE01	4	Poca experiencia con la arquitectura basada en EJB y Servicios Web que sean implementadas en el desarrollo del Proyecto.	Determinar las ventajas y desventajas sobre la implementación de una arquitectura orientada a servicios.

Fuente: Elaboración propia

Uno de los riesgos de implementar SOA recae en el conocimiento que se debe tener sobre los distintos procesos de negocio de la organización, ya que es necesario obtener sus funciones principales para que sean expuestas en diferentes capas de servicios. Para mitigar el riesgo RE01 se analizaron las ventajas y desventajas de una arquitectura basada en servicios para implementar los procesos de negocio de la Biblioteca de la Universidad Libre Sede Bosque Popular, que en este caso corresponden a la consulta de bibliografía por parte de la comunidad académica. Durante su adopción se tuvo en cuenta el bajo nivel de transferencia de datos involucrados en dicho proceso y la alta disponibilidad que se debe ofrecer al usuario.

Algunos de los beneficios que ofrece SOA:

- Favorece a la reutilización de código.
- Facilita la integración entre aplicaciones nuevas y existentes.

- El software generado basado en esta arquitectura es escalable y de fácil mantenimiento.

Un punto en contra de SOA:

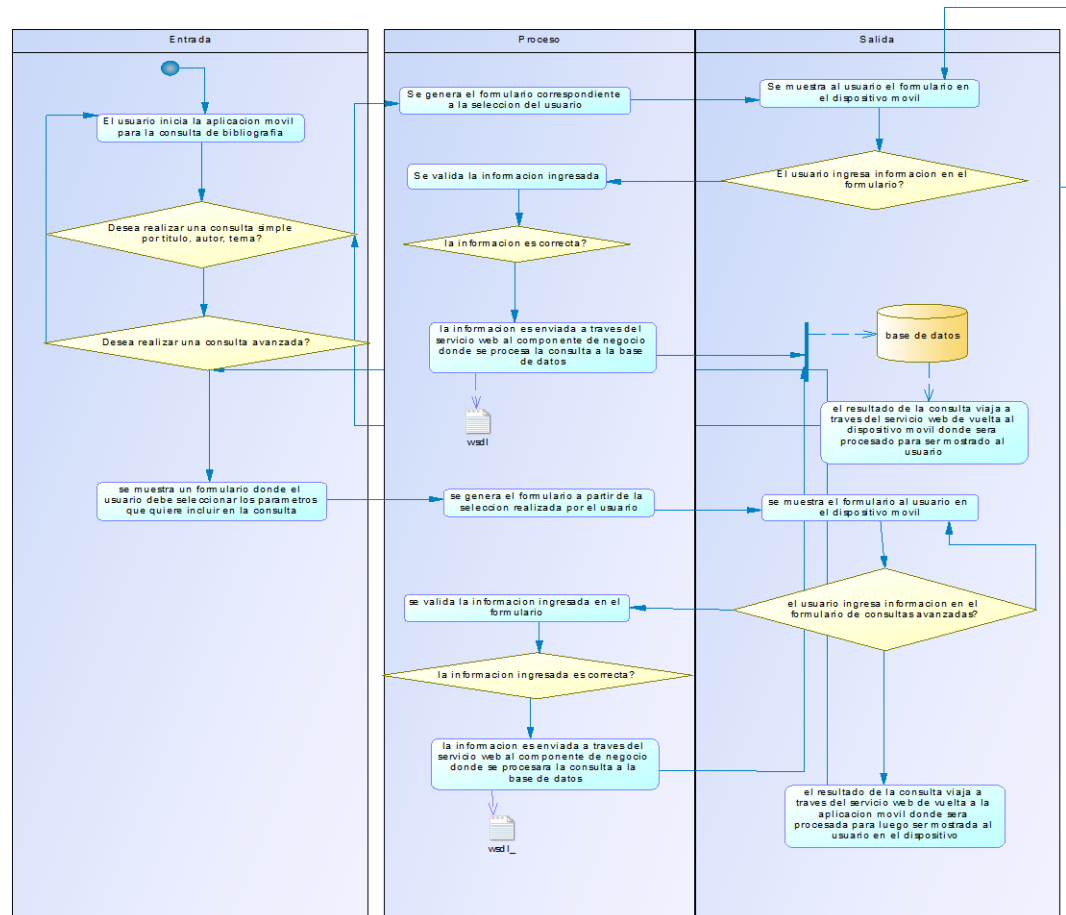
- SOA depende de la implementación de estándares de comunicación y sin ellos la interacción entre aplicaciones sería costosa en términos de tiempo y código.

Modelo de Negocio

La ilustración No. 11 representa el modelo de negocio del sistema propuesto que está compuesto por:

- Las entradas están representadas por la selección del tipo de consulta (por título, autor, tema o avanzada) realizada por el usuario.
- Los procesos están representados por la generación de los formularios solicitados por el usuario y la validación de la información ingresada en ellos posteriormente.
- Las salidas están representadas por el despliegue de los formularios en la pantalla del dispositivo móvil, las consultas realizadas a las bases de datos cuyos resultados serán mostrados en la pantalla del dispositivo

Ilustración 11 - Modelo de negocio del sistema propuesto



Fuente: Elaboración propia

Casos de Uso de consulta de bibliografía

A continuación se presentan las plantillas de casos de uso donde se describen paso a paso la interacción entre el usuario y el sistema propuesto.

La tabla No. 9 contiene la plantilla para el caso de uso de consulta de bibliografía por título. Este caso de uso consiste en la interacción del usuario con el sistema a través de la aplicación móvil, la cual muestra en pantalla un formulario con un campo donde deberá ser digitado el título del libro a buscar. El resultado de esta consulta será mostrado en una tabla compuesta por los registros que coinciden con el parámetro ingresado.

Tabla 9 - Caso de uso para la consulta de bibliografía por título

Nombre del Caso de Uso	Consultar bibliográfico por título
Código del Caso de Uso	1
Actor(es)	Usuario, Servicio Web, Aplicación Móvil, Aplicación Web.
Descripción	Este caso describe las acciones correspondientes a la consulta de bibliografía por título. El caso de uso inicia cuando el usuario ingresa el título del libro a buscar en el campo del formulario de búsqueda mostrado en la aplicación móvil y termina cuando la aplicación móvil procesa la respuesta generada por el servicio web para ser mostrada al usuario.
Precondición	1. La aplicación móvil deberá estar correctamente instalada en el dispositivo. 2. El teléfono móvil deberá estar conectado a la red inalámbrica de la Universidad, a la red EDGE o 3G. 3. El usuario debió haber ejecutado la aplicación móvil. 4. CU 5

	1. El usuario selecciona la opción de “Consulta por Título” en la lista que aparece en el menú inicial.	2. La aplicación móvil genera el formulario con el campo de texto donde el usuario digitará el título del libro a buscar y se mostrará en la pantalla del dispositivo
	3. El usuario digita el título del libro en el campo del formulario y ejecuta el comando “Buscar” ubicado en el menú.	4. La aplicación móvil se comunica con el servicio web, el cual envía una petición de consulta a la aplicación web con el parámetro ingresado por el usuario y con éste se realizará la consulta en la base de datos. El resultado de la consulta será retornado desde la aplicación web por el servicio web a la aplicación móvil donde se procesará la respuesta (CU 7).
Post condición	En el menú de la aplicación aparece el comando “Atrás”, que le permitirá al usuario volver al formulario de consulta.	
Flujo Excepcional	1. La aplicación mostrará un mensaje de alerta informando al usuario que se ha producido un error al realizar la consulta, luego se mostrará nuevamente el formulario de búsqueda.	2. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta no se puede realizar si ha dejado en blanco el campo de texto, luego se mostrará nuevamente el formulario de búsqueda.
	3. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta realizada no arroja resultado alguno, luego se mostrará nuevamente el formulario de búsqueda.	
Frecuencia	20 veces al día	

Importancia	Alta
Comentarios	Ninguno

Fuente: Elaboración propia

La tabla No. 10 contiene la plantilla para el caso de uso de consulta de bibliografía por autor. Este caso de uso consiste en la interacción del usuario con el sistema a través de la aplicación móvil, la cual muestra en pantalla un formulario con un campo donde deberá ser digitado el nombre del autor del material a buscar. El resultado de esta consulta será mostrado en una tabla compuesta por los registros que coinciden con el parámetro ingresado.

Tabla 10 - Caso de uso para la consulta de bibliografía por autor

Nombre del Caso de Uso	Consultar bibliografía por autor
Código del Caso de Uso	2
Actor(es)	Estudiante, Docente, Servicio Web, Aplicación Móvil, Aplicación Web
Descripción	Este caso de uso describe las acciones correspondientes a la consulta de bibliografía por autor. El caso de uso inicia cuando el usuario ingresa el nombre del autor del libro en el campo del formulario de búsqueda mostrado en la aplicación móvil y termina cuando la aplicación móvil procesa la respuesta generada por el servicio web para ser mostrada al usuario.
Precondición	1. La aplicación móvil deberá estar correctamente instalada en el dispositivo.

	2. El teléfono móvil deberá estar conectado a la red inalámbrica de la Universidad, a la red EDGE o 3G. 3. El usuario debió haber ejecutado la aplicación móvil. 4. CU 5	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario selecciona la opción de “Consulta por Autor” en la lista que aparece en el menú inicial.	2. La aplicación móvil genera el formulario con el campo de texto donde el usuario digitará el nombre del autor a buscar y se mostrará en la pantalla del dispositivo
	3. El usuario digita el nombre del autor en el campo del formulario y ejecuta el comando “Buscar” ubicado en el menú.	4. La aplicación móvil se comunica con el servicio web, el cual envía una petición de consulta a la aplicación web con el parámetro ingresado por el usuario y con éste se realizará la consulta en la base de datos. El resultado de la consulta será retornado desde la aplicación web por el servicio web a la aplicación móvil donde se procesará la respuesta (CU 7).
Post condición	En el menú de la aplicación aparece el comando “Atrás”, que le permitirá al usuario volver al formulario de consulta.	
Flujo Excepcional	1. La aplicación mostrará un mensaje de alerta informando al usuario que se ha producido un error al realizar la consulta, luego se mostrará nuevamente el	2. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta no se puede realizar si ha dejado en blanco el campo de texto, luego se mostrará nuevamente el

	formulario de búsqueda.	formulario de búsqueda.
	3. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta realizada no arroja resultado alguno, luego se mostrará nuevamente el formulario de búsqueda.	
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 11 contiene la plantilla para el caso de uso de consulta de bibliografía por tema. Este caso de uso consiste en la interacción del usuario con el sistema a través de una aplicación móvil, el cual muestra en pantalla un formulario con un campo donde deberá ser digitado el tema del libro a consultar. El resultado de la consulta será mostrado en una tabla compuesta por los registros que coinciden con el parámetro ingresado.

Tabla 11 - Caso de uso para la consulta de bibliografía por tema

Nombre del Caso de Uso	Consultar bibliografía por tema	
Código del Caso de Uso	3	
Actor(es)	Estudiante, Docente, Servicio Web, Aplicación Móvil	
Descripción	Este caso de uso describe las acciones correspondientes a la consulta de bibliografía por tema. El caso de uso inicia cuando el usuario ingresa el tema del libro a buscar en el campo del formulario de búsqueda mostrado en la aplicación móvil y termina cuando la aplicación móvil procesa la respuesta generada por el servicio web para ser mostrada al usuario.	
Precondición	<ol style="list-style-type: none"> 1. La aplicación móvil deberá estar correctamente instalada en el teléfono móvil. 2. El teléfono móvil deberá estar conectado a una red inalámbrica, a la red EDGE o 3G. 3. El usuario deberá haber ejecutado la aplicación móvil. 4. CU 5 	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario selecciona la opción de “Consulta por Tema” en la lista que aparece en el menú	2. La aplicación móvil genera el formulario con el campo de texto donde el usuario digitará

	inicial.	el nombre del autor a buscar y se mostrará en la pantalla del dispositivo
	3. El usuario digita el tema del libro en el campo del formulario y ejecuta el comando “Buscar” ubicado en el menú.	4. La aplicación móvil se comunica con el servicio web, el cual envía una petición de consulta a la aplicación web con el parámetro ingresado por el usuario y con éste se realizará la consulta en la base de datos. El resultado de la consulta será retornado desde la aplicación web por el servicio web a la aplicación móvil donde se procesará la respuesta (CU 7).
Post condición	En el menú de la aplicación aparece el comando “Atrás”, que le permitirá al usuario volver al formulario de consulta.	
Flujo Excepcional	1. La aplicación mostrará un mensaje de alerta informando al usuario que se ha producido un error al realizar la consulta, luego se mostrará nuevamente el formulario de búsqueda.	2. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta no se puede realizar si ha dejado en blanco el campo de texto, luego se mostrará nuevamente el formulario de búsqueda.
	3. La aplicación mostrará un mensaje de alerta informando al usuario que	

	la consulta realizada no arroja resultado alguno, luego se mostrará nuevamente el formulario de búsqueda.
Frecuencia	20 veces al día
Importancia	Alta
Comentarios	Ninguno

Fuente: Elaboración propia

La tabla No.12 contiene la plantilla para el caso de uso de consulta avanzada de bibliografía. Este caso de uso consiste en la interacción del usuario con el sistema a través de una aplicación móvil, la cual muestra en pantalla un formulario donde deberán ser seleccionados los parámetros a consultar. Posteriormente se muestra en pantalla el formulario con los campos correspondientes a la selección realizada donde se digitarán los datos. El resultado de la consulta será mostrado en una tabla compuesta por los registros que coinciden con los parámetros ingresados.

Tabla 12 - Caso de uso para consulta avanzada de bibliografía

Nombre del Caso de Uso	Consulta avanzada de bibliografía
Código del Caso de Uso	4
Actor(es)	Estudiante, Docente, Servicio Web, Aplicación Móvil
Descripción	Este Caso de Uso describe las acciones correspondientes a la Consulta de Bibliografía Avanzada. El caso de uso inicia cuando el usuario combina el título, autor o tema del libro a buscar e ingresa la información en los

	campos respectivos. El caso de uso termina cuando la aplicación móvil procesa la respuesta generada por el servicio web para ser mostrada al usuario.	
Precondición	1. La aplicación móvil deberá estar correctamente instalada en el dispositivo. 2. El teléfono móvil deberá estar conectado a una red inalámbrica, a la red EDGE o 3G. 3. El usuario deberá haber ejecutado la Aplicación Móvil. 4. CU 5	
Flujo Principal	Acción actor	Acción sistema
	1. El usuario selecciona la opción de “Consulta Avanzada” en la lista que aparece en el menú inicial.	2. La aplicación móvil genera el formulario con parámetros que se pueden agregar a la consulta (título, autor, tema) acompañados por un checkbox para que puedan ser seleccionados y se mostrará en la pantalla del dispositivo.
	3. El usuario por medio del teclado podrá seleccionar que campos desea agregar en la consulta avanzada. Para continuar deberá ejecutar el comando “Continuar” del menú	4. La aplicación móvil genera el formulario de consulta basado en la selección realizada por el usuario en el formulario y será mostrado en la pantalla del dispositivo
	5. El usuario digita la información correspondiente en los campos del formulario y ejecuta el comando “Buscar” ubicado en el menú.	6. La aplicación móvil se comunica con el servicio web, el cual envía una petición de consulta a la aplicación web con los parámetros ingresados por el usuario y con éste se realizará la consulta en la base de datos.

		El resultado de la consulta será retornado desde la aplicación web por el servicio web a la aplicación móvil donde se procesará la respuesta (CU 7).
Post condición	En el menú de la aplicación aparece el comando “Atrás”, que le permitirá al usuario volver al formulario de consulta.	
Flujo Excepcional	1. La aplicación mostrará un mensaje de alerta informando al usuario que se ha producido un error al realizar la consulta, luego se mostrará nuevamente el formulario de búsqueda.	2. La Aplicación mostrará un mensaje de Alerta informando al usuario que la consulta no se puede realizar si ha dejado en blanco uno de los campos de texto, luego se mostrará nuevamente el formulario de búsqueda.
	3. La aplicación mostrará un mensaje de alerta informando al usuario que la consulta realizada no arroja resultado alguno, luego se mostrará nuevamente el formulario de búsqueda.	4. La aplicación mostrará un mensaje de alerta si el usuario selecciona solo un campo de búsqueda. Se requiere que sean seleccionados mínimo dos.
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 13 contiene la plantilla para el caso de uso del login de usuario. Este caso de uso consiste en la interacción del usuario con el sistema a través de una aplicación móvil, la cual muestra en pantalla un formulario donde serán ingresados el nombre de usuario y contraseña proporcionados para el ingreso a la aplicación. Si los datos ingresados son correctos será mostrado al usuario el menú correspondiente a su rol, de lo contrario se mostrará un mensaje indicando que los datos suministrados son incorrectos.

Tabla 13 - Caso de uso para el login de usuario

Nombre del Caso de Uso	Login de usuario	
Código del Caso de Uso	5	
Actor(es)	Usuario, Servicio Web, Aplicación Móvil	
Descripción	Este caso describe las acciones correspondientes al ingreso del usuario a la aplicación. El caso de uso inicia cuando el usuario digita el nombre y contraseña en el formulario mostrado cuando se inicia la aplicación móvil y termina cuando se valida la información y el usuario accede al menú correspondiente.	
Precondición	<ol style="list-style-type: none"> 1. La aplicación móvil deberá estar correctamente instalada en el dispositivo. 2. El teléfono móvil deberá estar conectado a una red inalámbrica, a la red EDGE o 3G. 3. El usuario debió haber ejecutado la aplicación móvil. 	
Flujo Principal	Acción actor	Acción sistema
	<ol style="list-style-type: none"> 1. El usuario ejecuta la aplicación móvil instalada en el dispositivo. 	<ol style="list-style-type: none"> 2. El dispositivo móvil carga en memoria la aplicación y muestra el formulario de ingreso donde se deberán

		<p>digital el nombre de usuario y contraseña</p>
	<p>3. El usuario ingresa los datos y ejecuta el comando “Ingresar” ubicado en el menú.</p>	<p>4. La aplicación enviará una petición de login a la aplicación web a través del servicio web con los parámetros ingresados por el usuario. Se realiza la consulta en la base de datos, pasando como parámetro el nombre de usuario. Para finalizar, se ejecuta el CU 6.</p>
Post condición	Se mostrará el menú principal en la pantalla del dispositivo	
Flujo Excepcional	<p>1. Si se presenta un error de comunicación se presenta un mensaje de alerta al usuario.</p>	<p>2. Si las credenciales ingresadas por el usuario son incorrectas, se presenta un mensaje de alerta al usuario.</p>
	<p>3. Si el usuario deja en blanco los campos del formulario y ejecuta el comando “Ingresar”, se presenta un mensaje de alerta indicando que los campos son obligatorios</p>	
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 14 contiene la plantilla para el caso de uso de validación de rol del usuario. Este caso de uso consiste en la validación del rol luego del ingreso a la aplicación móvil con el fin de mostrar en la pantalla del dispositivo el menú con las acciones correspondientes.

Tabla 14 - Caso de uso para la validación del rol de usuario

Nombre del Caso de Uso	Validar rol de usuario	
Código del Caso de Uso	6	
Actor(es)	Usuario, Servicio Web, Aplicación Móvil	
Descripción	Este caso describe las acciones correspondientes para obtener el rol de usuario. El caso de uso inicia cuando el CU 5 retorna la información correspondiente al usuario que ingresa a la aplicación y termina cuando se envía la identificación al servicio web para obtener el rol y mostrar el menú correspondiente en la aplicación móvil.	
Precondición	1. CU 5	
Flujo Principal	Acción actor	Acción sistema
	1. Realiza la petición de ingreso a la aplicación	2. La aplicación luego de validar los datos ingresados por el usuario (CU 5) ahora procede a validar el rol asignado realizando una consulta a la base de datos. El servicio web retorna el código del rol a la aplicación móvil luego de la comunicación con la aplicación web. A

		continuación se genera el menú correspondiente
Post condición	Se mostrará el menú correspondiente en la pantalla del dispositivo	
Flujo Excepcional	1. Si se presenta un error de comunicación se presenta un mensaje de alerta al usuario.	2. Si el usuario no tiene un rol asignado se muestra un mensaje de alerta en la pantalla del dispositivo.
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 15 contiene la plantilla para el caso de uso de la transformación del resultado de la consulta. Este caso de uso consiste en la transformación del resultado de la consulta proveniente del servicio web que está almacenado en un arreglo que debe ser recorrido para obtener la información y pasarla a una estructura de datos específica para que pueda ser ubicada en una tabla que será mostrada en la pantalla del dispositivo.

Tabla 15 - Caso de uso para la transformación del resultado

Nombre del Caso de Uso	Transformar resultado	
Código del Caso de Uso	7	
Actor(es)	Servicio Web, Aplicación Móvil	
Descripción	Este caso describe las acciones correspondientes a la transformación del resultado de la consulta a la base de datos para que sea mostrada al usuario en la pantalla del dispositivo. El caso de uso inicia cuando el servicio web retorna el resultado de la consulta a la aplicación móvil en un arreglo de tipo objeto que debe ser recorrido para obtener la información y termina cuando el resultado es mostrado en una tabla en la pantalla del dispositivo.	
Precondición	CU 1, CU 2, CU 3, CU 4	
Flujo Principal	Acción actor	Acción sistema
	1. El servicio web retorna el resultado de la consulta y se ejecuta el CU 8	2. La aplicación móvil recorre el arreglo de tipo objeto para obtener los registros que serán almacenados en una estructura especial para que puedan ser mostrados en la

		pantalla del dispositivo móvil de forma tal que sea útil para el usuario ver las coincidencias obtenidas.
Post condición	El usuario podrá consultar la tabla que contiene los registros usando el teclado del dispositivo. La aplicación móvil estará lista para realizar otras consultas de bibliografía.	
Flujo Excepcional	1. Si el resultado de la consulta no contiene registros, se muestra un mensaje de advertencia al usuario indicando que no hay coincidencias.	
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

Fuente: Elaboración propia

La tabla No. 16 contiene la plantilla para el caso de uso de validación del resultado de la consulta. Este caso de uso consiste en la validación que se realiza al resultado cuando es retornado por el servicio web de tal manera que se notifique al usuario que no se encontraron coincidencias si dicho resultado no contiene registros

Tabla 16 - Caso de uso para la validación del resultado de la consulta

Nombre del Caso de Uso	Validar resultado consulta	
Código del Caso de Uso	8	
Actor(es)	Servicio Web, Aplicación Móvil	
Descripción	Este caso describe las acciones correspondientes a la validación del resultado obtenido en la consulta a la base de datos, que fue transportado desde la aplicación web por el servicio web hasta la aplicación móvil. El caso de uso inicia cuando el servicio web retorna el resultado de la consulta para ser validado en la aplicación móvil de forma tal que si no contiene registros se notifica al usuario y en caso contrario se continúa con el CU 7. El caso de uso termina cuando el resultado es validado.	
Precondición	CU 1, CU 2, CU 3, CU 4	
Flujo Principal	Acción actor	Acción sistema
	1. El servicio web retorna el resultado de la consulta que fue enviado desde la aplicación web	2. La aplicación móvil valida que la respuesta tenga registros. Si no contiene registro alguno, el usuario verá un aviso de advertencia

		en la pantalla del dispositivo. Si el resultado contiene más de un registro, se continúa con el CU 7
Post condición	El usuario podrá consultar la tabla que contiene los registros usando el teclado del dispositivo. La aplicación móvil estará lista para realizar otras consultas de bibliografía.	
Flujo Excepcional	1. Si el resultado de la consulta no contiene registros, se muestra un mensaje de advertencia al usuario indicando que no se encontraron coincidencias.	
Frecuencia	20 veces al día	
Importancia	Alta	
Comentarios	Ninguno	

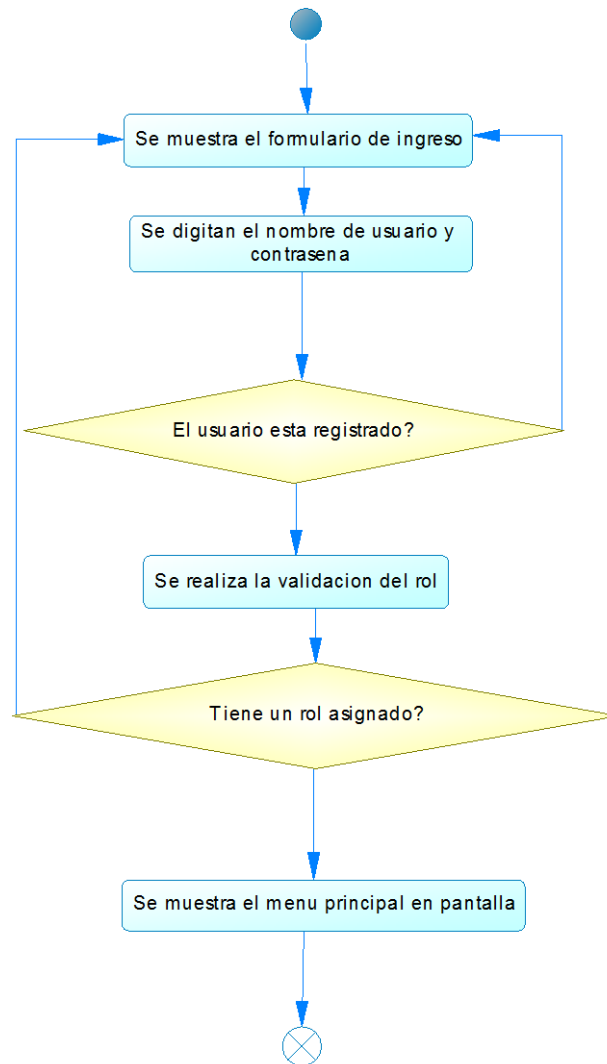
Fuente: Elaboración propia

Diagramas de actividad

Los diagramas de actividad son utilizados para representar el flujo de trabajo desde un punto inicial hasta uno final incluyendo las operaciones y las decisiones que se toman durante la actividad. A continuación se encuentran los diagramas de actividad relacionados con el sistema propuesto.

La ilustración No. 12 hace referencia al diagrama de actividad para el ingreso del usuario a la aplicación móvil.

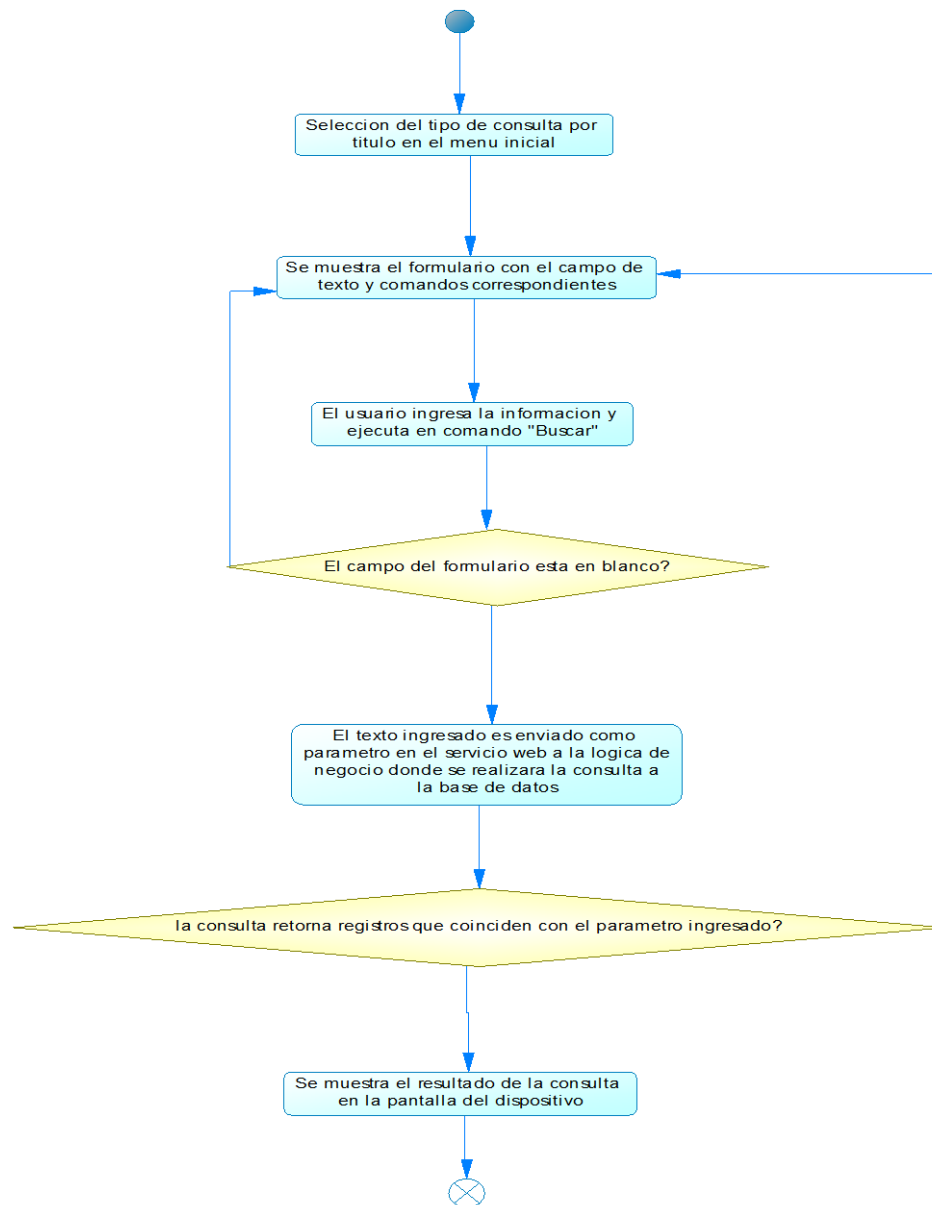
Ilustración 12 - Diagrama de actividad para el ingreso a la aplicación



Fuente: Elaboración propia

La ilustración No. 13 hace referencia al diagrama de actividad para la consulta de bibliografía por título.

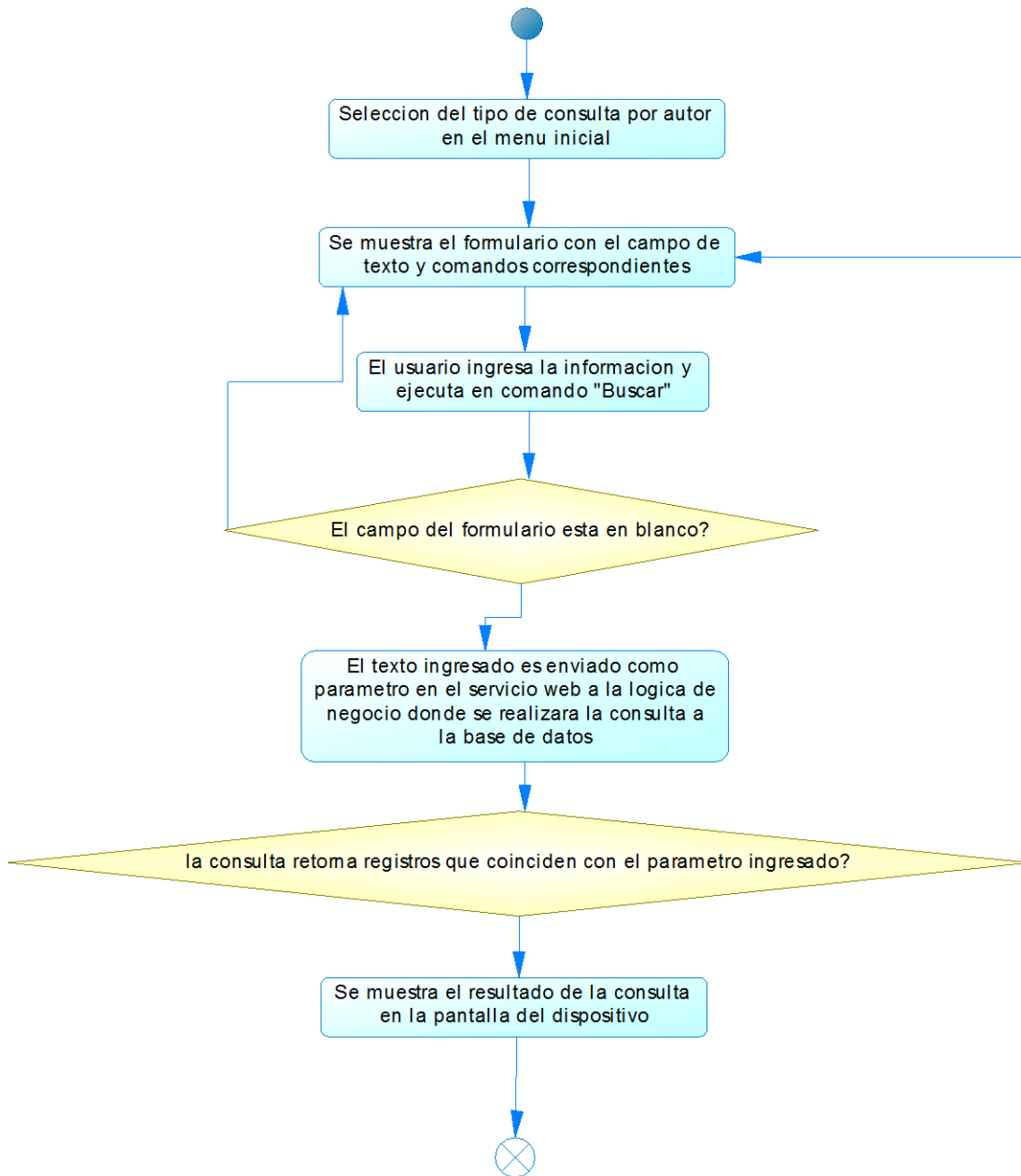
Ilustración 13 - Diagrama de actividad para la consulta de bibliografía por título



Fuente: Elaboración propia

La ilustración No. 14 hace referencia al diagrama de actividad para la consulta de bibliografía por autor.

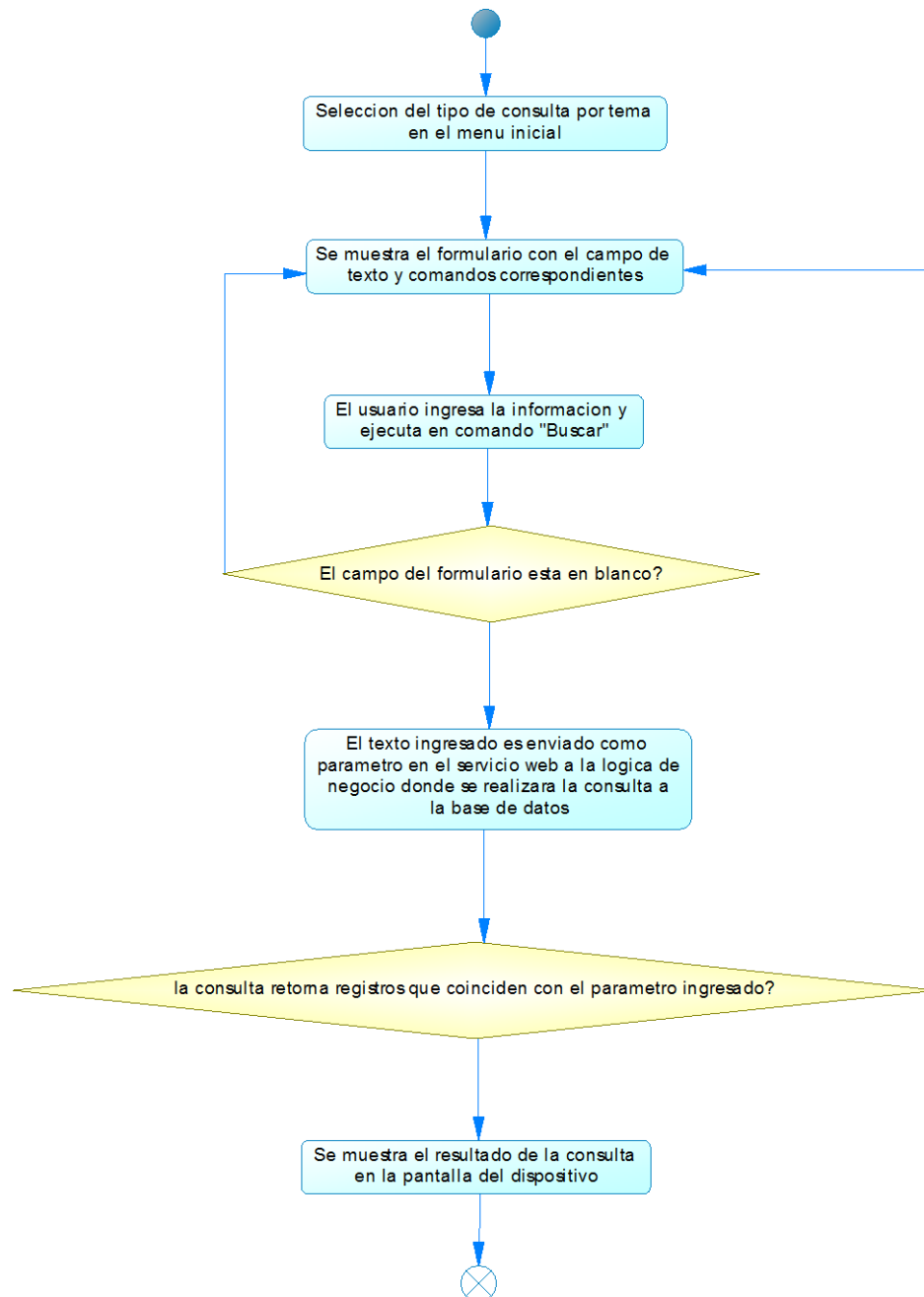
Ilustración 14 - Diagrama de actividad para la consulta de bibliografía por autor



Fuente: Elaboración propia

La ilustración No. 15 hace referencia al diagrama de actividad para la consulta de bibliografía por tema.

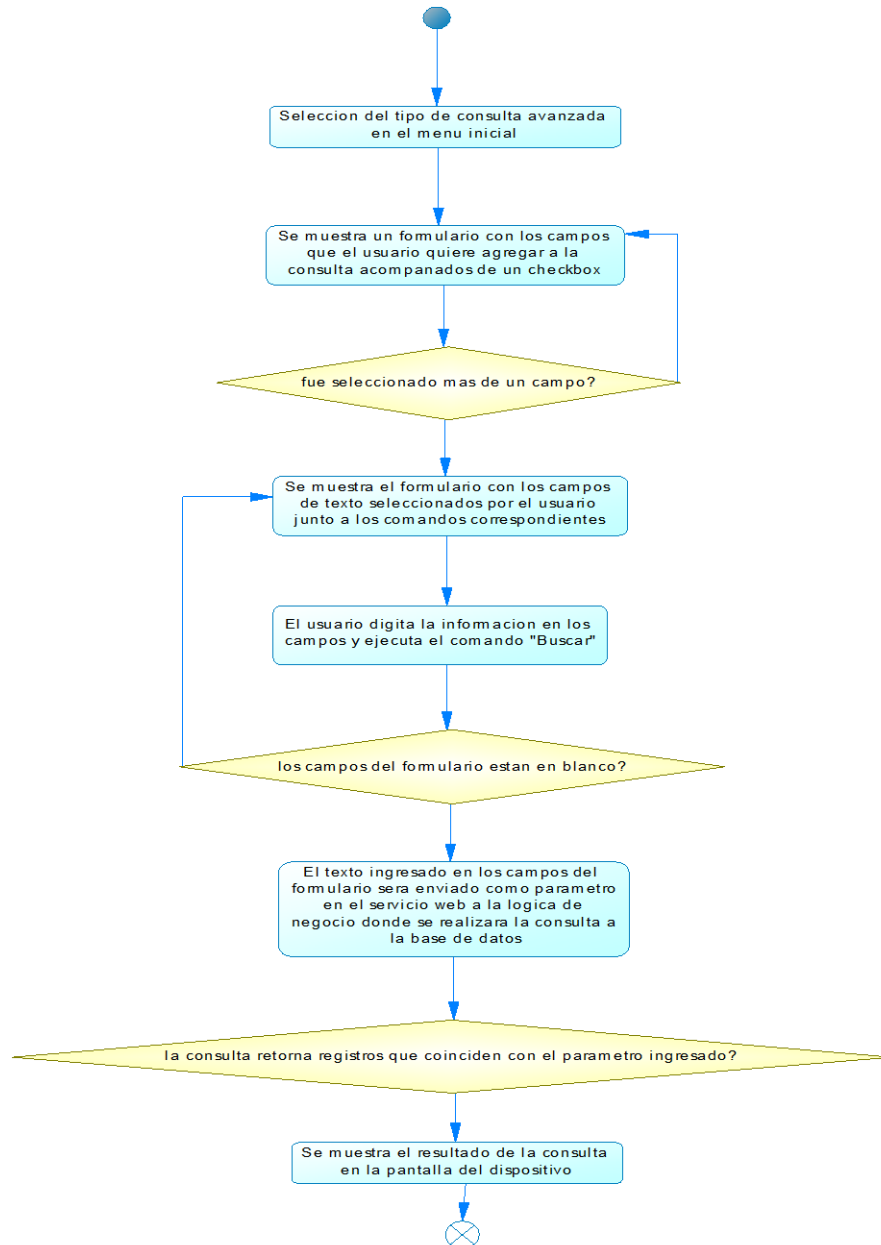
Ilustración 15 - Diagrama de actividad para la consulta de bibliografía por tema



Fuente: Elaboración propia

La ilustración No. 16 hace referencia al diagrama de actividad para la consulta avanzada de bibliografía.

Ilustración 16 - Diagrama de actividad para la consulta avanzada de bibliografía



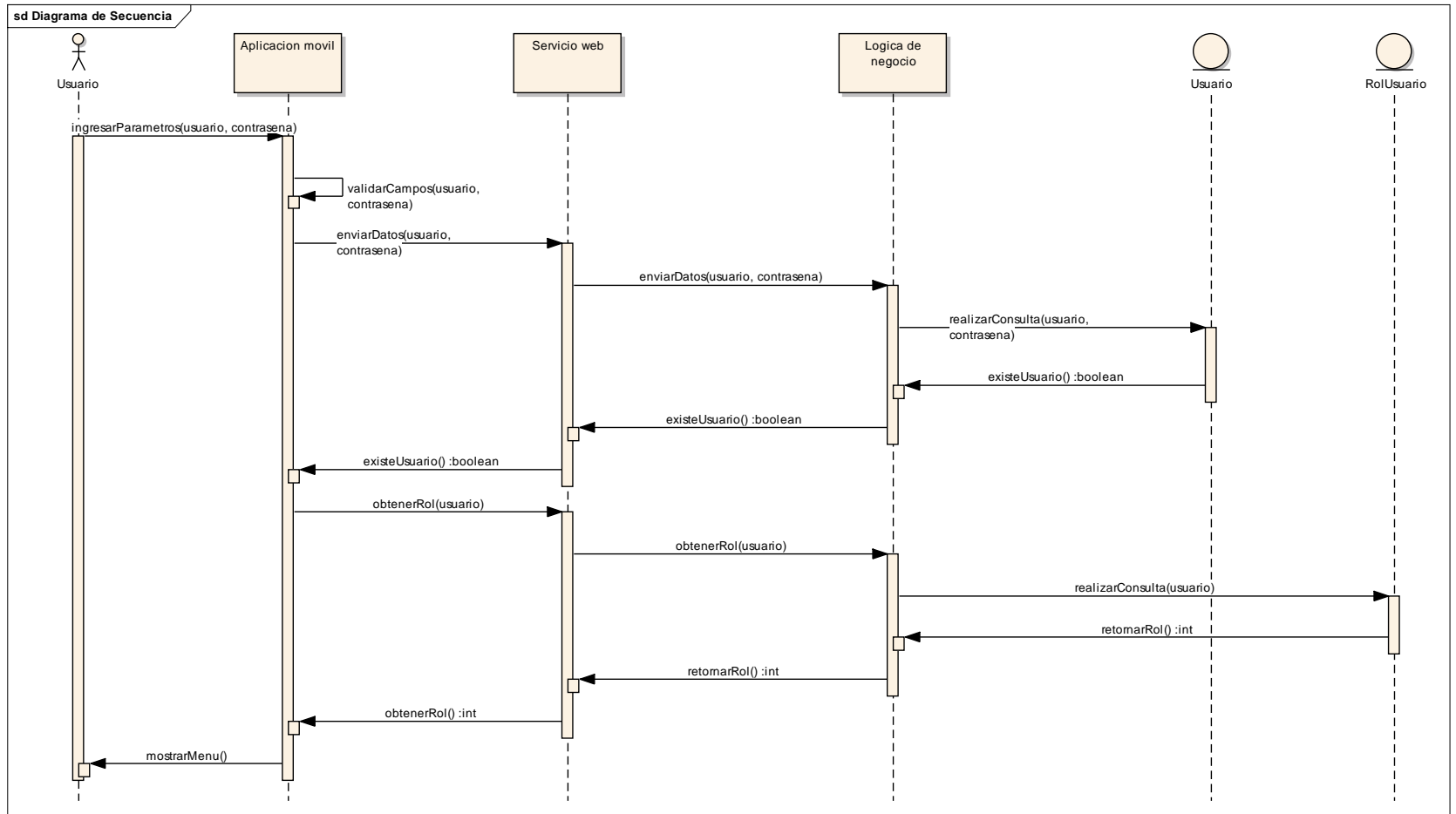
Fuente: Elaboración propia

Diagramas de secuencia

Los diagramas de secuencia representan la interacción de objetos representados en líneas de vida a través del tiempo por medio de mensajes, los cuales están compuestos por una serie de elementos como flechas que se dirigen desde la línea de vida origen hasta la línea de vida destino y representan el intercambio de mensajes. A continuación se realizará la descripción de los diagramas de secuencia involucrados.

La ilustración No. 17 hace referencia al diagrama de secuencia para el ingreso a la aplicación móvil que está compuesto por actores, líneas de vida, entidades y relaciones de interacción donde el usuario será el actor que interactúa con las líneas de vida *aplicación móvil*, *servicio web* y *lógica del negocio* por medio de mensajes enviados entre ellas. El actor usuario envía el mensaje “ingresarParámetros” con los parámetros correspondientes a las credenciales de ingreso a la línea de vida *aplicación móvil* donde se validan los campos del formulario y luego envía los datos a las líneas de vida *servicio web* y *lógica de negocio* que se encarga de enviar el mensaje “realizarConsulta” a la entidad Usuario que posteriormente retorna el mensaje “existeUsuario” con un valor de tipo booleano hasta la línea de vida *aplicación móvil*. A continuación, la línea de vida *aplicación móvil* envía el mensaje “obtenerRol” cuyo parámetro es el nombre de usuario y pasa por la línea de vida *servicio web* hasta la línea de vida *lógica de negocio* que envía el mensaje “realizarConsulta” a la entidad RolUsuario, la cual retorna el mensaje “retornarRol” con un valor de retorno de tipo entero hasta la línea de vida *aplicación móvil*. Para finalizar, la línea de vida *aplicación* retorna el mensaje “mostrarMenu” a la línea de vida del actor Usuario.

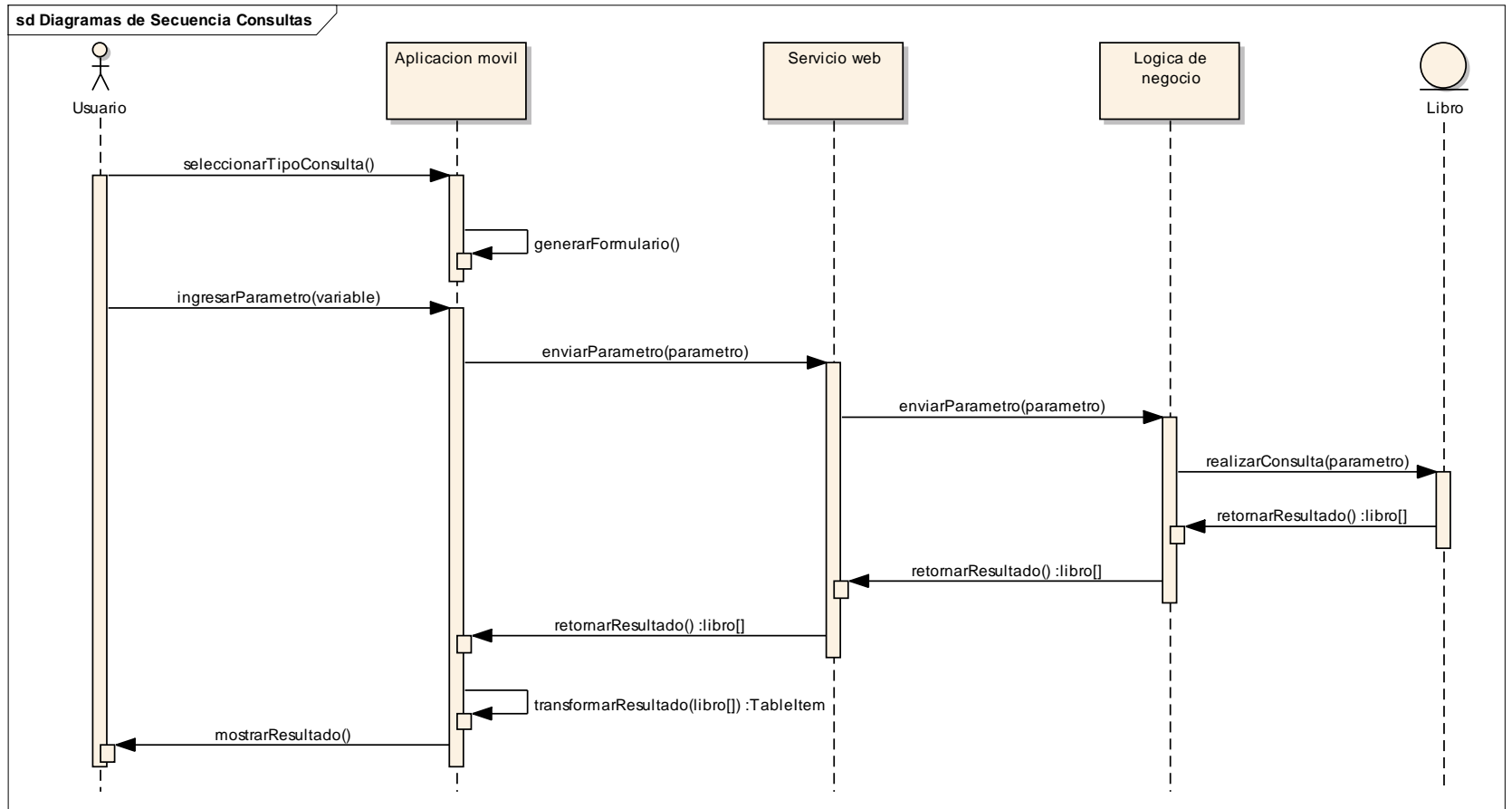
Ilustración 17 - Diagrama de secuencia para el ingreso del usuario a la aplicación móvil



Fuente: Elaboración propia

La ilustración No. 18 hace referencia al diagrama de secuencia para la consulta de bibliografía que está compuesto por actores, líneas de vida, entidades y relaciones de interacción donde el usuario será el actor que interactúa con las líneas de vida *aplicación móvil*, *servicio web* y *lógica del negocio* por medio de mensajes enviados entre ellas. El Usuario envía el mensaje “seleccionarTipoConsulta” a la línea de vida *aplicación móvil* y ella misma genera el mensaje “generarFormulario” que es retornado a sí misma. Luego, el usuario envía el mensaje “ingresarParámetro” con un parámetro de tipo carácter a las líneas de vida *servicio web* y *lógica de negocio* que posteriormente envía el mensaje “realizarConsulta” a la entidad *Libro*, la cual retorna el mensaje “retornarResultado” con un arreglo de tipo Libro hasta la línea de vida *aplicación móvil* donde se envía el mensaje “transformarResultado” a sí misma. Para finalizar, la línea de vida *aplicación móvil* envía el mensaje “mostrarResultado” a la línea de vida del actor *Usuario*.

Ilustración 18 - Diagrama de secuencia para la consulta de material



Fuente: Elaboración propia

3.1.3 Fase de Construcción

En el desarrollo de esta fase se da por terminado el análisis y el desarrollo de la solución propuesta. Ahora los esfuerzos se enfocan en la mitigación de los riesgos relacionados con la construcción del software y el desarrollo de la solución de acuerdo a lo descrito en la Fase de Elaboración. Con una descripción del sistema propuesto se busca enunciar cuales son las tecnologías utilizadas y su aplicación dentro de la arquitectura implementada. Dicha descripción se complementa con los diagramas de clases que se componen por los atributos y relaciones, y con el diagrama de componentes que se encarga de mostrar los componentes que se encuentran en cada capa de la arquitectura.

Riesgos detectados en la fase de construcción

La tabla No. 17 contiene los riesgos detectados durante el desarrollo de la fase de construcción, que son clasificados según su magnitud y cada uno debe estar acompañado de un plan de contingencia.

Tabla 17 - Riesgos en la fase de construcción

Número del Riesgo	Clasificación del Riesgo / Magnitud	Descripción del Riesgo & Impacto	Estrategia de Mitigación y/o Plan de Contingencia
RC01	4	No se tiene suficiente experiencia en el manejo de las librerías JAX-WS para el desarrollo de Servicios Web.	Adquirir documentación sobre el desarrollo de servicios web que implemente JAX-WS. Desarrollar aplicaciones que sirvan para poner a prueba los conceptos adquiridos.

RC02	4	No se poseen los conocimientos suficientes en el desarrollo de aplicaciones móviles que consumen Servicios Web con la tecnología J2ME de Java.	<p>Adquirir documentación sobre el desarrollo de aplicaciones móviles con la tecnología J2ME.</p> <p>Adquirir documentación sobre desarrollo de aplicaciones móviles que consumen Servicios Web desarrollados con las librerías de JAX-WS.</p> <p>Desarrollar aplicaciones que sirvan para poner a prueba los conceptos adquiridos.</p>
------	---	------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Elaboración propia

Para mitigar el riesgo RC01 fue necesario consultar diferentes páginas en internet con tutoriales y guías que facilitaron adquirir nuevos conocimientos sobre el uso de las tecnologías JAXB que permite obtener un esquema XML a partir de una clase Java y JAX-WS que permite la construcción de servicios web basados en el protocolo SOAP y las operaciones están definidas en el WSDL.

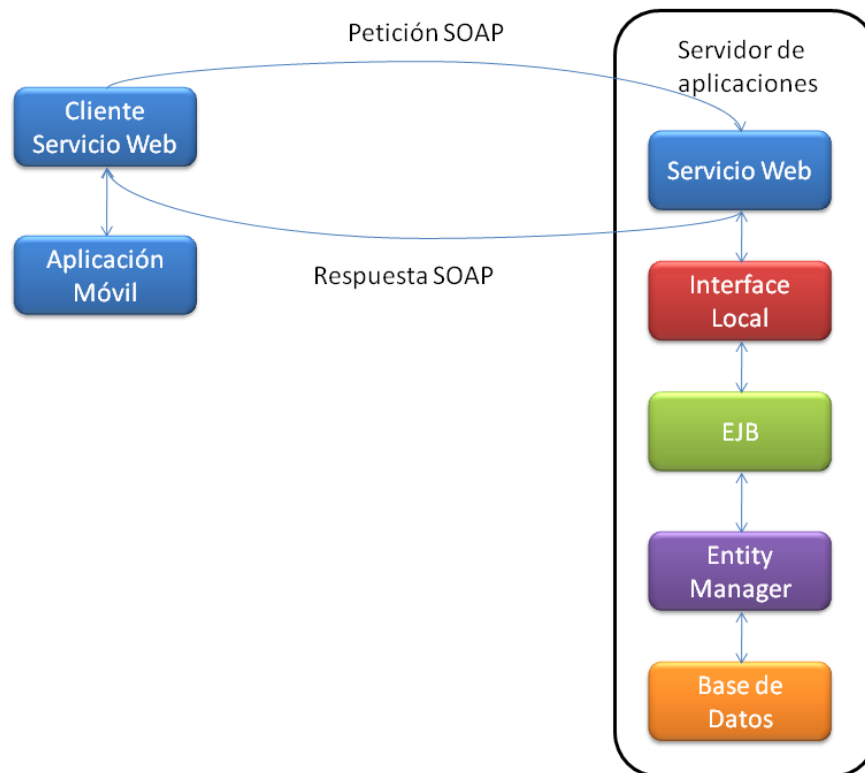
Para mitigar el riesgo RC02 se consultó el libro “J2ME Java 2 Micro Edition Manual de usuario y tutorial” que contiene varios ejemplos que sirvieron como base para la construcción de varios prototipos de la aplicación móvil. Adicionalmente se consultaron páginas en internet con tutoriales y guías relacionadas con el desarrollo de un cliente móvil desarrollado con la tecnología JME que consume un servicio web.

Descripción del Sistema Propuesto

El sistema propuesto está conformado por tres componentes fundamentales:

- Una aplicación móvil desarrollada con la tecnología Java ME que será instalada en dispositivos móviles compatibles con CLDC y MIDP.
- Una aplicación empresarial desarrollada con la tecnología Java EE compuesta por componentes de negocio EJB 3.0 y acceso a base de datos a través de JPA.
- Un servicio web y su respectivo cliente móvil que permite la comunicación entre los dos primeros por medio del protocolo SOAP.

Ilustración 19 - Arquitectura de la solución propuesta



Fuente: Elaboración propia

La ilustración No. 19 describe de forma general la arquitectura de la solución que está compuesta por distintas especificaciones pertenecientes a la versión 5 de la tecnología Java EE (Java Enterprise Edition), como EJB (Enterprise Java Beans), Servicios Web y JPA (Java Persistence Api). También se hace uso de la tecnología Java ME (Java Micro Edition) para el desarrollo de la aplicación móvil utilizada por los usuarios.

Los EJB (Enterprise Java Beans) versión 3.0 hacen parte de los componentes de negocio incluidos en la especificación de Java 5 y a comparación de las versiones anteriores, la versión la 3.0 requiere que el EBJ implemente una interface que bien puede ser local o remota. Este componente de negocio será utilizado junto a la interface “Entity Manager”, quien es la encargada de ejecutar distintas operaciones sobre las clases entidad. Dichas clases son la representación de las tablas almacenadas en la base de datos y están definidas en la “Unidad de Persistencia” junto a la conexión a la base de datos. En el proyecto se implementó un EJB de tipo “Stateless”, el cual se caracteriza por no mantener una conversación constante con el cliente, a comparación con el EJB “Stateful” que permitirá mantener en sesión dicha interacción, permitiendo el acceso a los métodos de consulta a través de peticiones efectuadas por el cliente para poder disponer de la información solicitada.

En los métodos establecidos para el aplicativo se hará uso de una instancia del “Entity Manager” con el objeto de permitir la ejecución de las diferentes consultas que se realizarán sobre las clases entidad. Como consecuencia de lo anterior, se obtendrá el material que se encuentra alojado en la base de datos y posteriormente será registrado dentro de un arreglo de tipo Libro, el cual es un objeto creado a partir de un “binding”. De acuerdo a la ilustración No. 20 se puede denotar dicho proceso, el cual es necesario para definir las clases diferentes a las nativas que posee Java (String, Double, Float, etc) que serán expuestas en el servicio web gracias al comando “schemagen”

que está incluido en las librerías de JAXB. Para ser mapeada, la clase no debe tener un método constructor y es necesario agregar unas anotaciones específicas para que el proceso ejecutado determine el control sobre la serialización de los atributos (XmlAccessorType) y el mapeo de la clase a un esquema XML (XmlType). El esquema resultante será utilizado por el servicio web como una referencia de los objetos que viajan durante el intercambio de mensajes con el cliente.

El asistente del entorno de desarrollo Netbeans solicita el ingreso del nombre que se le dará al “binding”, la ruta en el sistema de archivos donde está ubicado el esquema generado y por último el ingreso del nombre que se le dará al paquete donde serán creadas las clases dentro de la estructura del proyecto. Al finalizar, se podrán encontrar las clases Libro y ObjectFactory en el paquete asignado para que puedan ser utilizadas en los métodos del EJB.

Ilustración 20 - Proceso de generación del esquema

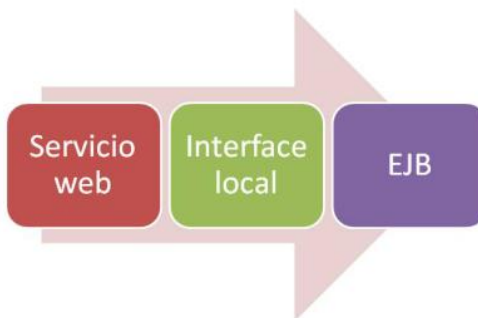


Fuente: Elaboración propia

Por medio del comando “wsimport”, que viene incluido en la instalación del JDK (Java Development Tool Kit), se generará la clase Libro y las interfaces que exponen los métodos del servicio web desplegado en el servidor de aplicaciones.

Un servicio web describe una forma estándar de integrar aplicaciones web usando los protocolos abiertos XML, SOAP y WSDL sobre el protocolo HTTP. XML es usado para etiquetar los datos, SOAP para la transferencia de datos y WSDL para describir los servicios disponibles. El entorno de desarrollo Netbeans también cuenta con un asistente que facilita la generación de un servicio web a partir de los métodos existentes en un EJB. Dicho asistente solicita el nombre que se le dará al servicio web, el nombre del paquete donde será alojada la clase y para concluir el proceso se debe seleccionar el EJB creado en el proyecto. El servicio web creado por el asistente está compuesto por varias anotaciones, entre ellas está la anotación “Web Service” sobre la definición de la clase y “Web method” sobre los métodos que lo componen. Cabe anotar que los métodos expuestos en el servicio web son los mismos que componen el EJB y para acceder a ellos será a través de una instancia de la interface local como se puede observar en la ilustración No. 21.

Ilustración 21 - Comunicación entre el servicio web y el Enterprise Java Bean



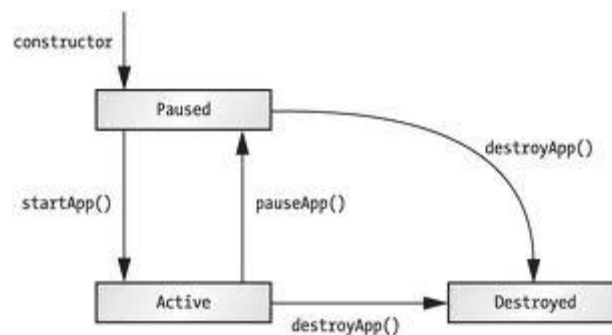
Fuente: Elaboración propia

Para el desarrollo de la aplicación móvil se hizo uso de la tecnología Java ME (Micro Edition), la cual se caracteriza por estar diseñada para dispositivos móviles que se caracterizan por tener una limitación en la capacidad de procesamiento y cantidad de memoria. Por este motivo, la configuración

CLDC (Connected Limited Device Configuration) 1.1 y el perfil MIDP (Mobile Information Device Profile) 2.0 son las más apropiadas para asegurar la compatibilidad de la aplicación con la mayoría de equipos que se encuentran en el mercado actual.

La aplicación móvil está compuesta por una clase principal que controla el ciclo de vida a través de los métodos “startApp”, “pauseApp” y “destroyApp” que son heredados de la clase MIDlet, la cual hace parte de la especificación JME (Java Micro Edition). Además se hará uso de clases que extienden la clase Form para el manejo de los formularios que serán mostrados en la pantalla del dispositivo y le permiten al usuario ingresar texto en campos de texto, ejecutar las consultas por medio de comandos asignados a teclas del dispositivo móvil y ver el resultado de las consultas.

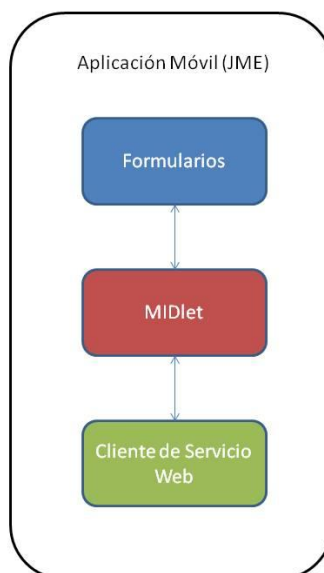
Ilustración 22 - Ciclo de vida de una aplicación móvil con la tecnología Java Micro Edition



Fuente: Tomada de http://www.digit.lk/j2me_march2009

La ilustración No. 22 representa las fases que componen el ciclo de vida de una aplicación móvil desarrollada con la tecnología Java Micro Edition.

Ilustración 23 - Arquitectura de la Aplicación Móvil



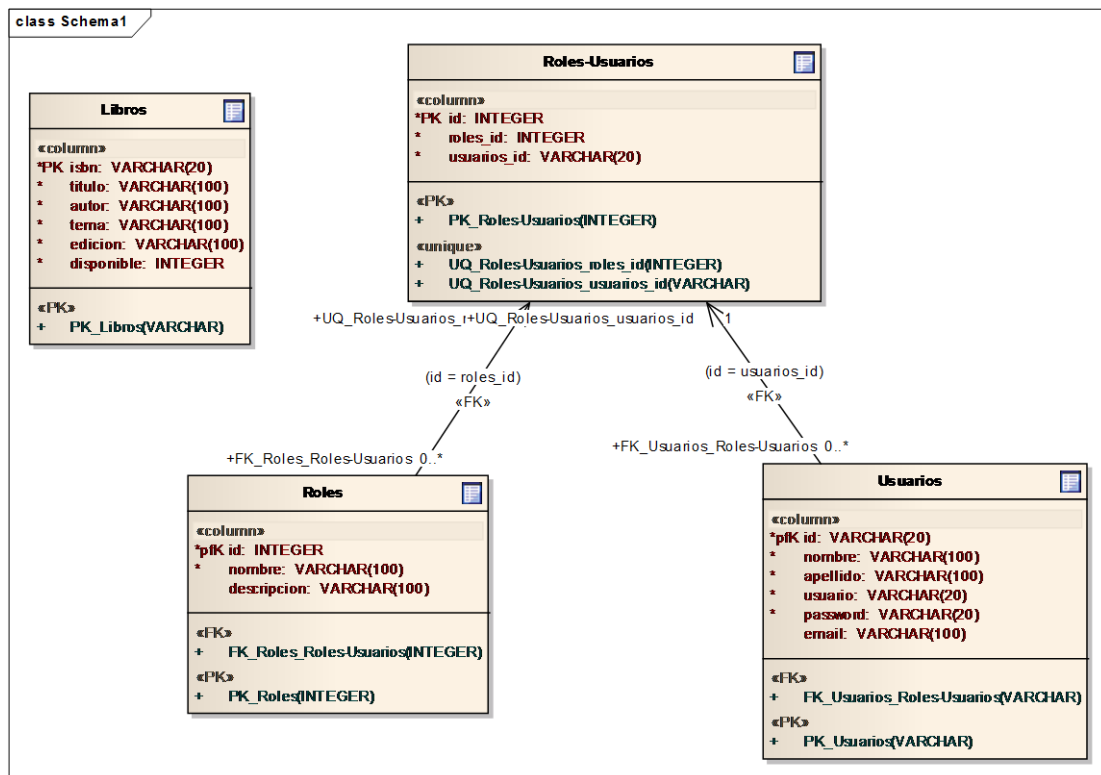
Fuente: Elaboración propia

La ilustración No. 23 representa la implementación de una arquitectura MVC (Modelo Vista Controlador) en la aplicación móvil desarrollada con la tecnología Java ME. La clase MIDlet se encarga del control en la aplicación al administrar todas las acciones que son iniciadas por el usuario desde los formularios y realizar la invocación de los métodos expuestos en el servicio a través del cliente del servicio web, mientras que la vista está conformada por las clases que representan los distintos formularios que son mostrados al usuario en la pantalla del dispositivo móvil.

Modelo de base de datos

La ilustración No. 24 representa el modelo de base de datos que contendrá la información que será utilizada por la aplicación.

Ilustración 24 - Modelo de base de datos



Fuente – Elaboración propia

A continuación se encuentra el diccionario de datos, el cual contiene el nombre, el tipo, el tamaño y la descripción de cada uno de los campos que hacen parte del modelo de datos

La tabla No. 18 contiene el diccionario de datos para la tabla “Libros”

Nombre de archivo: libros

Descripción: archivo con la información de los libros que se encuentran en la biblioteca.

Tabla 18 – Diccionario de datos para la tabla Libros

Campo	Tipo	Tamaño	Descripción
Isbn	VARCHAR	20	Identificador del libro
Titulo	VARCHAR	100	Título del libro
Autor	VARCHAR	100	Autor del libro
Tema	VARCHAR	100	Tema del libro
Edicion	VARCHAR	100	Edición del libro
Disponible	INT	3	Número de libros disponibles

Fuente – Elaboración propia

Relaciones: Ninguna

Campos clave: isbn

La tabla No. 19 contiene el diccionario de datos para la tabla “Usuarios”

Nombre de archivo: usuarios

Descripción: archivo con la información de los usuarios del sistema

Tabla 19 – Diccionario de datos para la tabla Usuarios

Campo	Tipo	Tamaño	Descripción
Id	VARCHAR	20	Identificación del usuario
Nombre	VARCHAR	100	Nombre del usuario
Apellido	VARCHAR	100	Apellido del usuario
Usuario	VARCHAR	20	Nombre de usuario para el ingreso a la aplicación
Password	VARCHAR	20	Contraseña para el ingreso a la aplicación
Email	VARCHAR	100	Correo electrónico del usuario

Fuente – Elaboración propia

Relaciones: Roles_Usuarios

Campos clave: Id

La tabla No. 20 contiene el diccionario de datos para la tabla “Roles”

Nombre de archivo: roles

Descripción: archivo con la información del rol de cada usuario

Tabla 20 – Diccionario de datos para la tabla Roles

Campo	Tipo	Tamaño	Descripción
Id	INTEGER	3	Identificador del rol
Nombre	VARCHAR	100	Nombre del rol
Descripción	VARCHAR	100	Descripción del rol

Fuente – Elaboración propia

Relaciones: Roles_Usuarios

Campos clave: Id

La tabla No. 21 contiene el diccionario de datos para la tabla “Roles Usuarios”

Nombre de archivo: roles_usuarios

Descripción: archivo con la información de los roles por usuario

Tabla 21 - Diccionario de datos para la tabla Roles Usuarios

Campo	Tipo	Tamaño	Descripción
RolesId	INTEGER	3	Identificación del rol
UsuariosId	VARCHAR	20	Identificación del usuario

Fuente – Elaboración propia

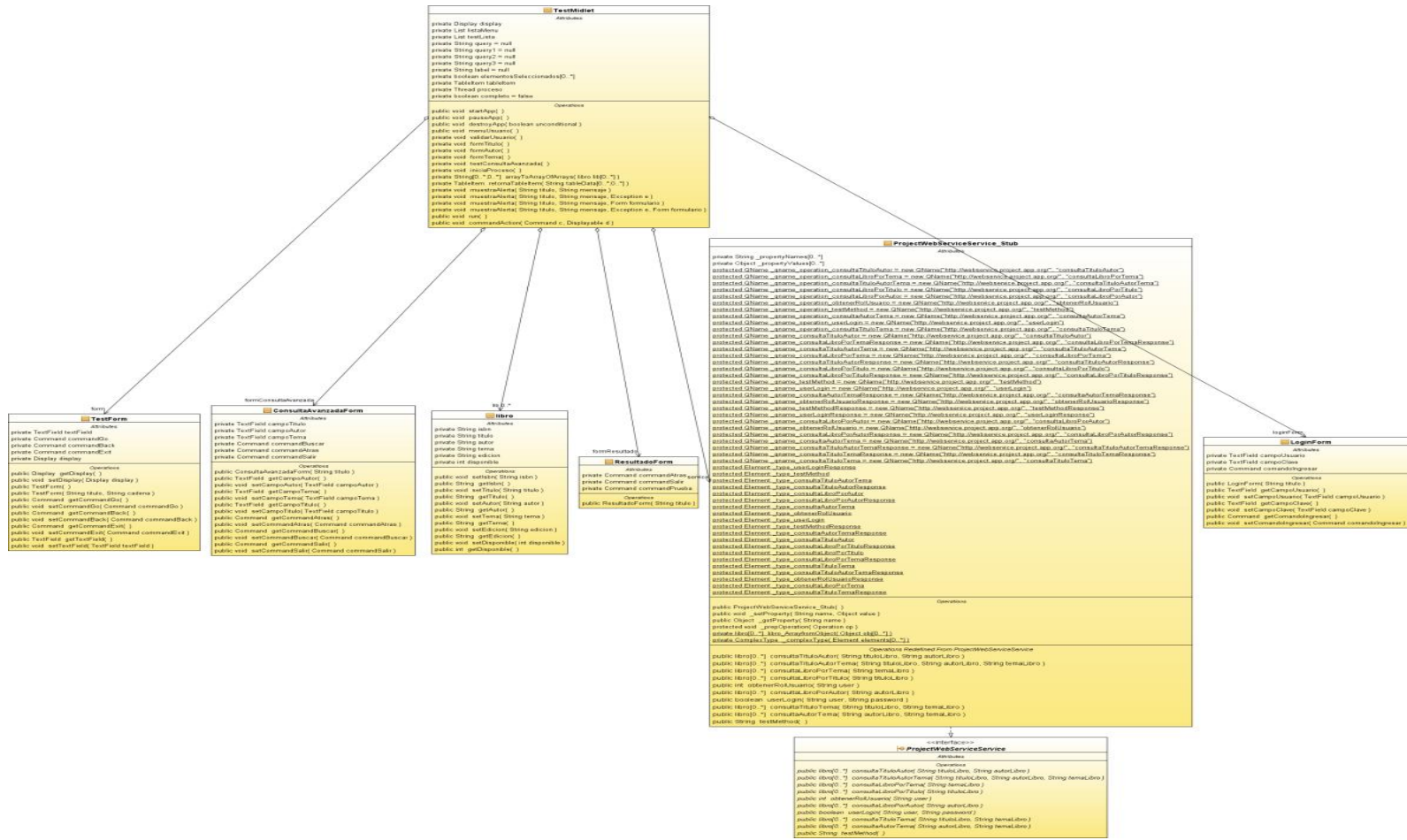
Relaciones: Roles, Usuarios

Campos clave: RolesId, UsuariosId

Diagramas de clase

Los diagramas de clase son fundamentales dentro del análisis y diseño de la aplicación propuesta, ya que estos representan la estructura de un sistema compuesto por clases (incluyendo sus tipos), las cuales son abstracciones de objetos compuestas por atributos y operaciones, con sus contenidos, relaciones estructurales y de herencia.

Ilustración 25 - Diagrama de clases para la aplicación móvil



Fuente: Elaboración propia

La ilustración No. 24 hace referencia a las clases que conforman la aplicación móvil y en los siguientes párrafos se hará una breve descripción de los métodos que hacen parte de ellas.

Clase TestMidlet

La clase TestMidlet es pública, de modo que otras clases pertenecientes a otros paquetes pueden tener acceso a ella. Adicionalmente hereda de la clase MIDlet los métodos encargados de manejar el ciclo de vida de la aplicación: startApp, stopApp, pauseApp y destroyApp. Realiza la implementación de la clase CommandListener para manejar los eventos disparados por los comandos de la aplicación y de la clase Runnable que permite el manejo de hilos para la comunicación con el servicio web sin que afecte el rendimiento de la aplicación móvil. TestMidlet está compuesta por los siguientes métodos:

- El método startApp es público, no recibe parámetros y no tiene un valor de retorno. Será el primer método que se ejecuta cada vez que es iniciada la aplicación y se encarga de obtener el display donde se mostrarán todos los formularios de la aplicación.
- El método pauseApp es público, no recibe parámetros y no tiene un valor de retorno. Se encarga de realizar alguna acción mientras la aplicación está pausada al momento de recibir una llamada o se consulta un mensaje de texto.
- El método destroyApp es público, no tiene un valor de retorno pero si recibe un parámetro de tipo booleano. Al invocar este método, la aplicación móvil llegará al final del ciclo de vida.
- El método menuUsuario es público, no recibe parámetros y no tiene un valor de retorno. Se encarga de crear una nueva instancia de una lista a la cual se le agregarán los tipos de búsqueda que el usuario podrá realizar luego del ingreso a la aplicación. Para finalizar, a la lista

se le agrega el método `commandListener` que se encarga de disparar eventos con el método `commandListener`.

- El método `validarUsuario` es público, no recibe parámetros y no tiene un valor de retorno. Se encarga de invocar el método `iniciaProceso` donde se dará inicio al hilo que realizará la conexión al servicio web y de esta forma se evita bloquear la aplicación móvil si se llega a presentar un problema.
- El método `formTitulo` es de tipo privado ya que será usado únicamente por la clase `TestMidlet`, no recibe parámetros y no tiene un valor de retorno. El método se encarga de crear una nueva instancia de la clase `TestForm` y su método constructor recibe dos parámetros de tipo `String` para generar el formulario correspondiente. Al formulario generado es agregada la funcionalidad encargada de disparar eventos con el método `commandListener` y se mostrará en la pantalla del dispositivo.
- El método `formAutor` es de tipo privado ya que será usado únicamente por la clase `TestMidlet`, no recibe parámetros y no tiene valor de retorno. El método se encarga de crear una nueva instancia de la clase `TestForm` y su método constructor recibe dos parámetros de tipo `String` para generar el formulario correspondiente. Al formulario generado es agregada la funcionalidad encargada de disparar eventos con el método `commandListener` y se mostrará en la pantalla del dispositivo.
- El método `formTema` es de tipo privado ya que será usado únicamente por la clase `TestMidlet`, no recibe parámetros y no tiene valor de retorno. El método se encarga de crear una nueva instancia de la clase `TestForm` y su método constructor recibe dos parámetros de tipo `String` para generar el formulario correspondiente. Al formulario generado es agregada la funcionalidad encargada de disparar eventos

con el método `commandListener` y se mostrará en la pantalla del dispositivo.

- El método `testConsultaAvanzada` es de tipo privado ya que será usado únicamente por la clase `TestMidlet`, no recibe parámetros y no tiene un valor de retorno. En el método se crea una nueva instancia de una lista múltiple que le permite al usuario seleccionar más de una opción relacionada con el tipo de consulta, se crean instancias de comandos y se agregan a la lista. Para terminar, a la lista le es agregada la funcionalidad encargada de disparar eventos con el método `commandListener` y se mostrará en la pantalla del dispositivo.
- El método `iniciaProceso` es de tipo privado porque será usado únicamente por la clase `TestMidlet`, no recibe parámetros y no tiene un valor de retorno. Este método se encarga de crear una nueva instancia de la clase `Thread` para ser iniciado.
- El método `arrayToArrayOfArrays` es de tipo privado porque será usado únicamente por la clase `TestMidlet`, recibe como parámetro un arreglo de tipo libro y retorna un arreglo de arreglos de tipo `String`. El arreglo de tipo libro recibido en el método es retornado por el servicio web luego del proceso de consulta y su contenido es almacenado en un arreglo de arreglos de tipo `String`.
- El método `retornaTableItem` es de tipo privado porque será usado únicamente por la clase `TestMidlet`, recibe como parámetro un arreglo de arreglos de tipo `String` y retorna un objeto de tipo `TableItem`. Dentro del método se crea un arreglo de tipo `String` con los títulos que llevarán las columnas de la tabla de resultados y se crea una instancia de la clase `SimpleTableModel` con el arreglo de arreglos de tipo `String` y el arreglo con las columnas. Para finalizar, se crea una instancia de la clase `TableItem` con el objeto de tipo `Display`, un título para la tabla y el modelo que contiene el resultado de la consulta para ser retornada y mostrada en la pantalla del dispositivo.

- El método `muestraAlerta` es de tipo privado porque será usado únicamente por la clase `TestMidlet` y es el único con sobrecarga, lo que permite declarar el mismo método que recibe diferentes parámetros. El método se encarga de crear una instancia de la clase `Alert` y luego es asignada a la instancia de la clase de tipo `Display` para mostrar mensajes de alerta en la pantalla del dispositivo cuando se quiere notificar al usuario sobre un error o advertencia.
- El método `run` es público y puede ser usado por cualquier clase dentro de la aplicación además no recibe ni retorna parámetros. Este método debe ser implementado debido al uso de la interface `Runnable` y se encarga de ejecutar el llamado al servicio web en un proceso paralelo o hilo para prevenir el bloqueo de la aplicación móvil si se llega a presentar un error de comunicación.
- El método `commandAction` es público y por esto puede ser usado por cualquier clase dentro de la aplicación. Recibe un parámetro de tipo `Command`, el cual contiene la información del comando que fue ejecutado y a partir de esta se ejecuta la acción correspondiente. También se recibe un parámetro de tipo `Displayable` pero no es usado.

Clase `TestForm`

La clase `TestForm` es pública, de forma que pueda ser instanciada en otras clases para tener acceso a sus atributos y métodos. Los atributos de esta clase son de tipo `TextField` y `Command` y son usados para crear los formularios que serán mostrados al usuario luego de la selección del tipo de consulta de bibliografía. `TestForm` está compuesta por los siguientes métodos:

- El método `TestForm` es el constructor de la clase y recibe dos parámetros de tipo `String` que son el título y una cadena de texto que

luego será evaluada para determinar el tipo de formulario que debe ser mostrado al usuario.

- Los métodos “getters” y “setters” para cada atributo de la clase que se encargan de acceder y modificar dichos atributos.

Clase ConsultaAvanzadaForm

La clase ConsultaAvanzadaForm es pública y por esto puede ser instanciada en cualquier clase para acceder a sus atributos y métodos. Además hereda la clase Form, lo cual permite que los atributos de tipo TextField y Command puedan ser agregados al formulario para ser mostrados en la pantalla del dispositivo cuando el usuario seleccione la opción de realizar la consulta avanzada. La clase ConsultaAvanzadaForm tiene los siguientes métodos:

- El método constructor ConsultaAvanzadaForm es de tipo público y recibe el parámetro título de tipo String, cuya función es poner el título del formulario que será mostrado en la pantalla del dispositivo. En el cuerpo del método se crean las instancias de los comandos y se agregan al formulario.
- Los métodos “getters” y “setters” para cada atributo de la clase y su función en acceder y modificar dichos atributos.

Clase ResultadoForm

La clase ResultadoForm es de tipo público para que pueda ser instanciada por cualquier clase y así tener acceso a sus propiedades y atributos. ResultadoForm hereda la clase Form, lo que permite que sus atributos puedan ser agregados en el formulario que será mostrado al usuario en la pantalla del dispositivo luego de la consulta de bibliografía. La clase ResultadoForm está compuesta por los siguientes métodos:

- El método constructor ResultadoForm es público y recibe el parámetro título, de tipo String, que se encarga de poner el título al formulario que será mostrado en la pantalla del dispositivo. En el cuerpo del método serán instanciados los atributos de tipo Command y luego son agregados al formulario.

Clase LoginForm

La clase LoginForm es pública, lo que permite que pueda ser instanciada en otras clases y se pueda acceder a sus atributos y métodos. LoginForm hereda la clase Form, lo que permite que los atributos de tipo TextField y Command sean agregados al formulario que será mostrado cuando el usuario ejecute la aplicación móvil. La clase LoginForm está compuesta por los siguientes métodos:

- El método LoginForm es el constructor y es de tipo público. Recibe como parámetro el título que es de tipo String y se encarga del título del formulario. En el cuerpo del método se instancian los objetos de tipo TextField y Command para ser agregados al formulario que será mostrado al usuario.
- Los métodos “getters” y “setters” que se encargan del acceso y modificación de los atributos pertenecientes a la clase.

Clase ProjectWebServiceService

La clase ProjectWebServiceService es una interface pública, lo que permite que sea implementada por otras clases. Esta clase fue generada gracias al asistente de creación de un cliente móvil de un servicio web del entorno de desarrollo integrado Netbeans. Los métodos de esta interface son los mismos que están expuestos en la clase que hace parte de la lógica del negocio que se encuentra alojada en el servidor de aplicaciones Glassfish. Los métodos serán descritos a continuación:

- El método `consultaTituloAutor` es público y recibe los parámetros `tituloLibro` y `autorLibro` que son de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `consultaTituloAutorTema` es público y recibe los parámetros `tituloLibro`, `autorLibro` y `temaLibro` que son de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `consultaLibroPorTema` es público y recibe el parámetro `temaLibro` que es de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `consultaLibroPorTitulo` es público y recibe el parámetro `tituloLibro` que es de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `obtenerRolUsuario` es público y recibe el parámetro `user` que es de tipo `String`. El método retorna un valor entero que representa el rol del usuario que está ingresando a la aplicación.
- El método `consultaLibroPorAutor` es público y recibe el parámetro `autorLibro` que es de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `userLogin` es público y recibe los parámetros `user` y `password` que son de tipo `String`. El método retorna un valor booleano que indica si el usuario que ingresa a la aplicación está o no registrado en la base de datos.
- El método `consultaTituloTema` es público y recibe los parámetros `tituloLibro` y `temaLibro` que son de tipo `String`. El método retorna un arreglo de tipo `libro`, el cual contiene el resultado de la consulta de bibliografía.
- El método `consultaAutorTema` es público y recibe los parámetros `autorLibro` y `temaLibro` que son de tipo `String`. El método retorna un

arreglo de tipo libro, el cual contiene el resultado de la consulta de bibliografía.

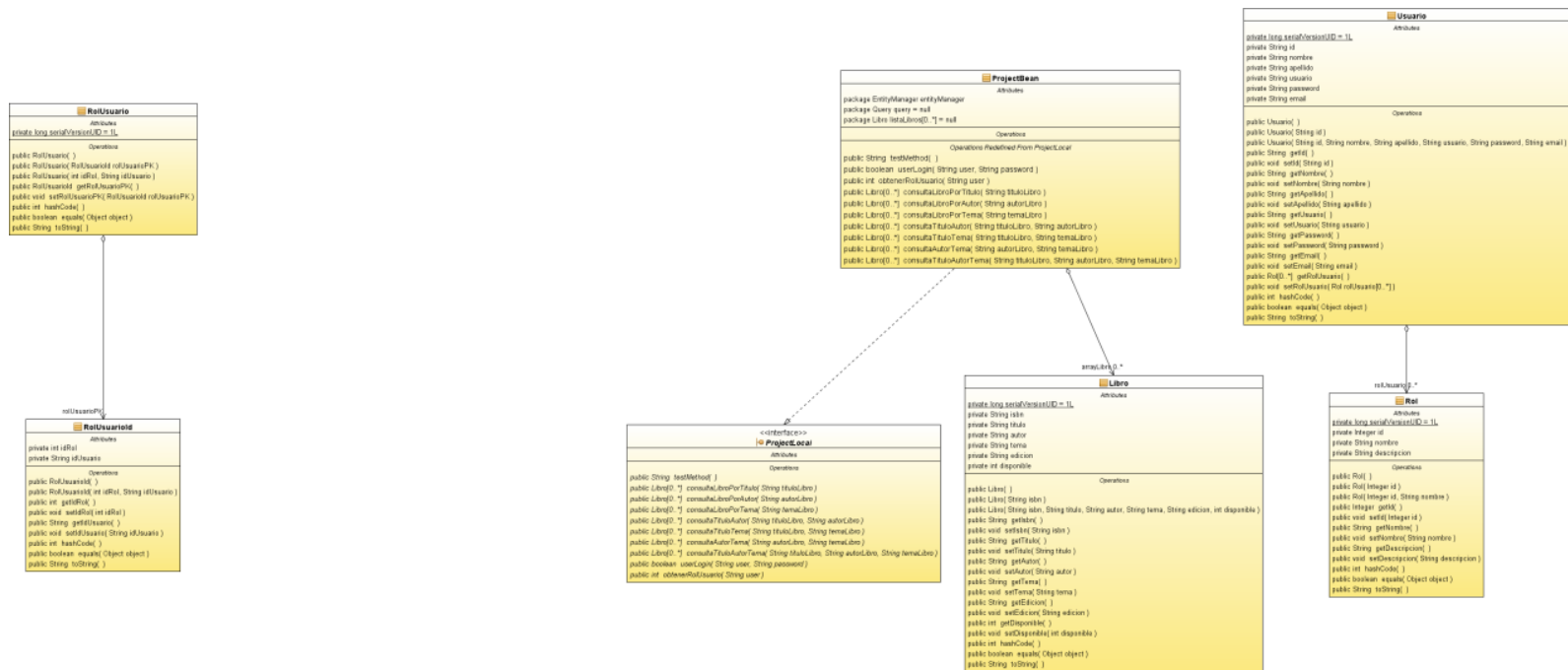
Clase ProjectWebServiceService_Stub

La clase ProjectWebServiceService_Stub es pública lo cual permite que pueda ser instanciada por otras clases. Además implementa las interfaces ProjectWebServiceService y javax.xml.rpc.Stub que obligan a la clase a implementar todos los métodos que hacen parte de ellas. Cabe anotar que esta clase fue generada automáticamente por el asistente de creación del cliente móvil perteneciente al entorno de desarrollo Netbeans, por lo cual se cuenta con la implementación de todos los métodos y con otro código necesario para su funcionamiento. Se realizó una corrección en el método libro_ArrayfromObject que es privado, estático, recibe un arreglo de tipo Object y retorna un arreglo de tipo Libro, que consiste en la asignación de la primera posición del arreglo que recibe el método en un nuevo arreglo de tipo Object y su longitud es requerida para tener una instancia de un arreglo de tipo Libro que será retornada al finalizar.

Clase libro

La clase Libro es pública, lo cual permite que pueda ser instanciada por otras clases y fue generada por el asistente de creación del cliente móvil perteneciente al entorno de desarrollo Netbeans. La clase libro corresponde a la clase que fue sometida al proceso de binding durante el desarrollo de la lógica de negocio que se encuentra desplegada en el servidor de aplicaciones. La clase tiene los siguientes atributos: isbn, título, autor, tema y edición que son de tipo String y el atributo disponible es de tipo entero. Los métodos de la clase corresponden a los respectivos getters y setters que se encargan del acceso y modificación de los atributos que pertenecen a la clase.

Ilustración 26 - Diagrama de clases para la lógica de negocio



Fuente: Elaboración propia

La ilustración No. 25 hace referencia al diagrama de clases que hacen parte de la lógica del negocio donde se realizan las diferentes consultas. A continuación se hará una breve descripción de los métodos que la componen.

Clase ProjectBean

La clase ProjectBean es un EJB (Enterprise Java Bean) que implementa la interface ProjectLocal para hacer uso de los métodos que realizan las diferentes consultas de bibliografía. Está compuesta por siguientes métodos:

- El método userLogin es público, lo que permite que pueda ser usado por otra clase y recibe dos parámetros de tipo String que son el nombre de usuario y la contraseña ingresados en el formulario de ingreso en la aplicación móvil y retornara un booleano. En el cuerpo del método será usada la instancia del EntityManager para realizar la consulta sobre la entidad Usuario y será retornado un objeto de este tipo. Si la contraseña ingresada es igual al atributo contraseña de la clase Usuario se retorna true o de lo contrario un false.
- El método obtenerRolUsuario es público, lo que permite su uso por otras clases y recibe como parámetro el nombre de usuario que ingresa a la aplicación que es de tipo String y retorna un valor de tipo int. En el cuerpo del método se realizan las consultas para obtener el rol del usuario por medio de la instancia del EntityManager. Si el usuario tiene asociado un rol entonces será retornado el identificador correspondiente, de lo contrario será retornado un cero.
- El método consultaLibroPorTitulo es público, lo que permite su uso por otras clases. Recibe como parámetro el título del libro ingresado en el formulario que se muestra en la aplicación móvil que es de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza la consulta sobre la entidad Libro, la cual retorna una lista que contiene objetos de tipo Libro y será recorrida por un Iterator con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo Libro que al final será retornado.
- El método consultaLibroPorAutor es público, lo que permite su uso por otras clases. Recibe como parámetro el autor del libro que fue

ingresado en el formulario de consulta en la aplicación móvil que es de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza la consulta sobre la entidad Libro, la cual retorna una lista que contiene objetos de tipo Libro y será recorrida por un Iterator con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo Libro que al final será retornado.

- El método `consultaLibroPorTema` es público, lo que permite su uso por otras clases. Recibe como parámetro el tema del libro, de tipo String, que fue ingresado en el formulario de consulta en la aplicación móvil y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza la consulta sobre la entidad Libro, la cual retorna una lista que contiene objetos de tipo Libro y será recorrida por un Iterator con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo Libro que al final será retornado.
- El método `consultaTituloAutor` es público, lo que permite su uso por otras clases. Recibe como parámetros el título y el autor del libro, ambos de tipo String, que fueron ingresados en el formulario de consulta avanzada y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza la consulta sobre la entidad Libro, la cual retorna una lista que contiene objetos de tipo Libro y será recorrida por un Iterator con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo Libro que al final será retornado.
- El método `consultaTituloTema` es público, lo que permite su uso por otras clases. Recibe como parámetros el título y el autor del libro, ambos de tipo String, que fueron ingresados en el formulario de consulta avanzada y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza la consulta sobre la entidad Libro, la cual retorna una lista que contiene objetos de tipo Libro y será recorrida por un Iterator con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo Libro que al final será retornado.

- El método `consultaAutorTema` es público, lo que permite su uso por otras clases. Recibe como parámetros el autor y el tema del libro, ambos de tipo `String`, que fueron ingresados en el formulario de consulta avanzada y retorna un arreglo de tipo `Libro`. En el cuerpo del método se realiza la consulta sobre la entidad `Libro`, la cual retorna una lista que contiene objetos de tipo `Libro` y será recorrida por un `Iterator` con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo `Libro` que al final será retornado.
- El método `consultaTituloAutorTema` es público, lo que permite su uso por otras clases. Recibe como parámetros el título, el autor y el tema del libro, todos de tipo `String`, que fueron ingresados en el formulario de consulta avanzada y retorna un arreglo de tipo `Libro`. En el cuerpo del método se realiza la consulta sobre la entidad `Libro`, la cual retorna una lista que contiene objetos de tipo `Libro` y será recorrida por un `Iterator` con el fin de obtener dichos objetos para ser almacenados en un arreglo de tipo `Libro` que al final será retornado.

Clase `ProjectLocal`

La clase `ProjectLocal` es una interface pública y es implementada por la clase `ProjectBean` que de acuerdo a la versión 3.0 de EJB (Enterprise Java Bean) debe implementar una interface local o remota. Esta clase contiene la definición de los métodos que deben ser implementados en la lógica de negocio:

- El método `consultaLibroPorTitulo` recibe un parámetro de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `consultaLibroPorAutor` recibe un parámetro de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `consultaLibroPorTema` recibe un parámetro de tipo `String` y retorna un arreglo de tipo `Libro`.

- El método `consultaTituloAutor` recibe dos parámetros de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `consultaTituloTema` recibe dos parámetros de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `consultaAutorTema` recibe dos parámetros de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `consultaTituloAutorTema` recibe tres parámetros de tipo `String` y retorna un arreglo de tipo `Libro`.
- El método `userLogin` recibe dos parámetros de tipo `String` y retorna un valor de tipo booleano.
- El método `obtenerRolUsuario` recibe un parámetro de tipo `String` y retorna un valor de tipo entero.

Clase Libro

La clase `Libro` es pública lo cual permite que pueda ser instanciada por cualquier clase y fue generada por el proceso de binding para hacer uso de ella durante la comunicación entre la aplicación móvil y la lógica de negocio expuesta en el servicio web. La clase tiene los atributos `isbn`, `título`, `autor`, `tema`, `edición` de tipo `String` y el atributo `disponible` de tipo entero. Los métodos de la clase corresponden a los getters y setters para cada atributo para realizar las operaciones de acceso y modificación.

Clase Usuario

La clase `Usuario` es pública lo cual permite que pueda ser instanciada por cualquier clase y fue generada por el asistente de creación de clases entidad del entorno de desarrollo Netbeans. Esta clase representa a la tabla “usuarios” que está almacenada en el motor de base de datos y se identifica como una clase entidad por la anotación `@Entity`. Los campos de la tabla

son representados como atributos de tipo String: id, nombre, apellido, usuario, password y email mientras que el atributo rolUsuario representa la relación con la tabla intermedia “rol_usuario”. Los métodos de la clase corresponden a los getters y setters que se encargan del acceso y modificación de los atributos.

Clase Rol

La clase Rol es pública lo cual permite que pueda ser instanciada por cualquier clase y fue generada por el asistente de creación de clases entidad del entorno de desarrollo Netbeans. Esta clase representa a la tabla “roles” que está almacenada en el motor de base de datos y se identifica como una clase entidad por la anotación @Entity. Los campos de la tabla son representados como atributos de tipo entero: id y los atributos de tipo String: nombre, descripción. Los métodos de la clase corresponden a los getters y setters que se encargan del acceso y modificación de los atributos.

Clase RolUsuario

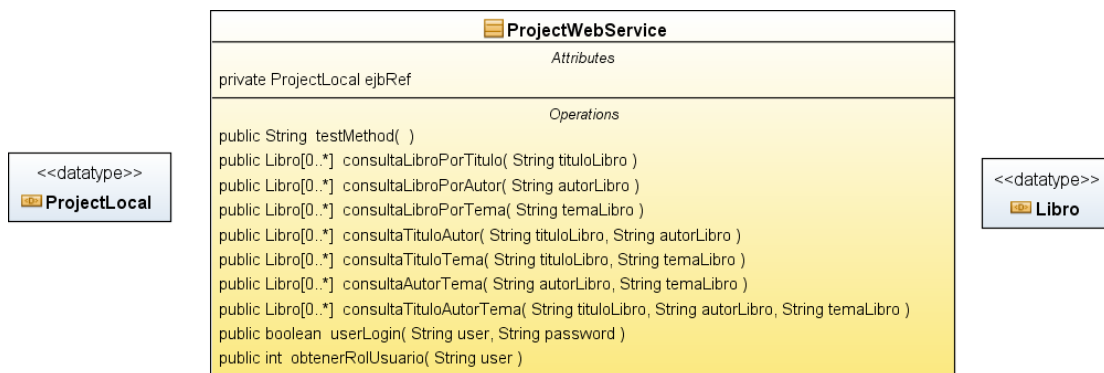
La clase RolUsuario es pública lo cual permite que pueda ser instanciada por cualquier clase y fue generada por el asistente de creación de clases entidad del entorno de desarrollo Netbeans. Esta clase representa a la tabla “roles_usuario” la cual es una tabla intermedia entre las tablas “roles” y “usuarios” y está almacenada en el motor de base de datos. Su único atributo corresponde a la llave primaria compuesta representada por la clase RolUsuariold y está acompañado por los métodos getters y setters que se encargan de su acceso y modificación.

Clase RolUsuariold

La clase RolUsuariold es pública lo cual permite que pueda ser instanciada por cualquier clase y fue generada por el asistente de creación de clases

entidad del entorno de desarrollo Netbeans. Esta clase representa una llave primaria compuesta por la llaves primarias de las tablas “roles” y “usuarios” que está almacenada en el motor de base de datos y se identifica como tal por la anotación @Embeddable. Los campos de la tabla son representados como atributos de tipo entero: idRol y de tipo String: idUsuario. Los métodos de la clase corresponden a los getters y setters que se encargan del acceso y modificación de los atributos.

Ilustración 27 - Diagrama de clases para la aplicación web



Fuente: Elaboración propia

La ilustración No. 26 hace referencia a las clases que hacen parte del servicio web que expone los métodos pertenecientes al EJB (Enterprise Java Bean) y se aloja en el servidor de aplicaciones Glassfish.

Clase ProjectWebService

La clase ProjectWebService es pública de modo que puede ser usada por otras clases. Los métodos que contiene serán descritos a continuación:

- El método `consultaLibroPorTitulo` es público, lo que permite que sea usado por otras clases. Recibe como parámetro el título del libro que

es de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaLibroPorTitulo del EJB a través de la interface local.

- El método consultaLibroPorAutor es público, lo que permite que sea usado por otras clases. Recibe como parámetro el autor del libro que es de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaLibroPorAutor del EJB a través de la interface local.
- El método consultaLibroPorTema es público, lo que permite que sea usado por otras clases. Recibe como parámetro el tema del libro que es de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaLibroPorTema del EJB a través de la interface local.
- El método consultaTituloAutor es público, lo que permite que sea usado por otras clases. Recibe como parámetros el título y el autor del libro que son de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaTituloAutor del EJB a través de la interface local.
- El método consultaTituloTema es público, lo que permite que sea usado por otras clases. Recibe como parámetros el título y el tema del libro que son de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaTituloTema del EJB a través de la interface local.
- El método consultaAutorTema es público, lo que permite que sea usado por otras clases. Recibe como parámetros el autor y el tema del libro que son de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaAutorTema del EJB a través de la interface local.
- El método consultaTituloAutorTema es público, lo que permite que sea usado por otras clases. Recibe como parámetros el título, el autor y el

tema que son de tipo String y retorna un arreglo de tipo Libro. En el cuerpo del método se realiza el llamado al método consultaTituloAutorTema del EJB a través de la interface local.

- El método userLogin es público, lo que permite que sea usado por otras clases a través de la instancia de la clase. Recibe como parámetro el nombre de usuario y contraseña que son de tipo String y retorna un valor booleano. En el cuerpo del método se realiza el llamado al método userLogin del EJB a través de la interface local.
- El método obtenerRolUsuario es público, lo que permite que sea usado por otras clases a través de la instancia de la clase. Recibe como parámetro el nombre de usuario que es de tipo String y retorna un valor de tipo int. En el cuerpo del método se realiza el llamado al método obtenerRolUsuario del EJB a través de la interface local.

El código fuente desarrollado en este proyecto se puede consultar en el anexo D.

Diagramas de componentes

Los diagramas de componentes representan como un sistema de software está dividido en partes físicas reemplazables y su objetivo es mostrar las relaciones entre ellas.

La ilustración No. 28 corresponde al diagrama de componentes correspondiente a las aplicaciones móvil y web. La aplicación móvil está conformada por los siguientes componentes:

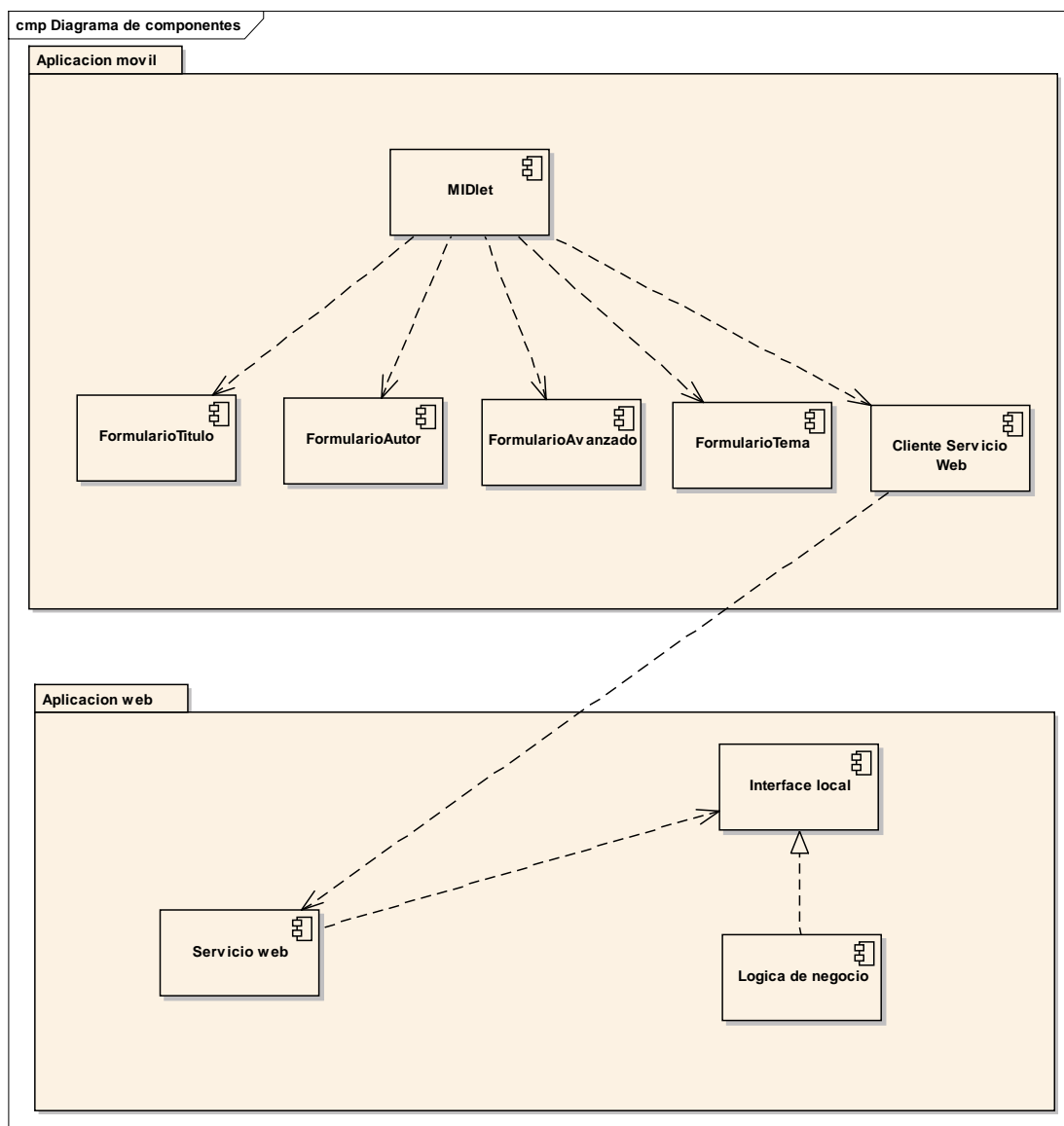
- El componente MIDlet representa a la aplicación móvil que se instala en el dispositivo y está asociada con los componentes FomularioLogin, FormularioTitulo, FormularioAutor, FormularioAvanzado y FormularioRespuesta que representan los formularios que el usuario visualiza en la pantalla del dispositivo. El

componente Cliente Servicio Web cumple con la función de conectar a la aplicación móvil con el componente Servicio Web que pertenece a la aplicación web.

La aplicación web está conformada por los siguientes componentes:

- En el componente Lógica de Negocio se realizan las operaciones de consulta a la base de datos para obtener la información correspondiente a las consultas de bibliografía que están definidas en el componente Interface Local. El componente Servicio Web se encarga de exponer las operaciones de consulta para que sea utilizado por el componente Cliente Servicio Web perteneciente a la aplicación móvil.

Ilustración 28 - Diagrama de componentes



Fuente: Elaboración propia

3.1.4 Fase de Transición

En la fase de transición se asegurará que el software que se entrega a los usuarios sea confiable, funcional y con un rendimiento aceptable de acuerdo a los resultados de las pruebas aplicadas sobre la versión obtenida en la fase de Construcción. Inicialmente se realiza la mitigación de los riesgos asociados a la metodología a implementar para el proceso de pruebas y a continuación se realiza la descripción del despliegue de las aplicaciones móvil y web en el ambiente de pruebas correspondiente. A continuación se generará un conjunto de pruebas que estará conformado por los módulos de casos de uso, operaciones con la base de datos, rendimiento, seguridad y configuración con el fin de verificar que el total de funcionalidades que componen el software operen correctamente. Esto será de gran utilidad para detectar inconsistencias que implicarán realizar ajustes al código fuente siempre y cuando se mantenga la funcionalidad establecida por la arquitectura.

Para finalizar será desarrollado el manual de usuario que hace parte de la documentación que será entregada junto al software implementado. Podrán ser consultados en el anexo E.

Riesgos detectados en la fase de transición

La tabla No. 22 contiene los riesgos detectados durante el desarrollo de la fase de inicio, que son clasificados según su magnitud y cada uno debe estar acompañado de un plan de contingencia.

Tabla 22 - Riesgos en la fase de transición

Número de Riesgo	Clasificación del Riesgo / Magnitud	Descripción del Riesgo & Impacto	Estrategia de Mitigación y/o Plan de Contingencia
RT01	4	Dificultad para elegir una o varias técnicas de pruebas de software.	Evaluar las diferentes técnicas para efectuar las pruebas del software desarrollado

Fuente: Elaboración propia

Se realizó una consulta sobre las diferentes técnicas para realizar pruebas sobre el desarrollo de software orientado a objetos y se organizaron en la tabla No. 23. Para este proceso se eligió la técnica llamada “prueba de escenarios de uso”, para la cual se generará un deck de pruebas (Anexo F) compuesto por los módulos de base de datos, funcionalidad o casos de uso, interface de usuario, rendimiento, seguridad y configuración. Cada uno de estos módulos tiene una descripción de la prueba, las precondiciones, los pasos necesarios, el resultado esperado, los hallazgos, la solución y su estado.

Para realizar las pruebas de funcionalidad se eligió la técnica llamada “revisión de prototipos” que consiste en generar un prototipo completo de la aplicación y sobre ésta se realizarán las pruebas a cada caso de uso. Durante el desarrollo de software se aplicarán las siguientes técnicas para garantizar que la solución cumple con las especificaciones propuestas por medio del “debug” o recorrido paso a paso que ofrece el entorno de desarrollo NetBeans:

- Prueba de caja negra
- Prueba de clases
- Prueba de integración de clases
- Prueba de integración
- Prueba de método
- Prueba de caminos
- Prueba de caja blanca

Tabla 23 - Técnicas para las pruebas de software orientado a objetos

Técnica	Descripción
Prueba de Caja-Negra	La prueba verifica que el ítem que se está probando cuando se dan las entradas apropiadas produce los resultados esperados.
Prueba de Valores-Frontera	Es la prueba de situaciones extremas o inusuales que el ítem debe ser capaz de manejar.
Prueba de Clases	Es el acto de asegurar que una clase y todas sus instancias cumplen con el comportamiento definido.
Prueba de Integración de Clases	Es el acto de asegurar que las clases, y sus instancias, conforman un software que cumple con el comportamiento definido.
Revisión de Código	Una forma de revisión técnica del código fuente que se entrega.
Prueba de Componente	Es el acto de validar que un componente funciona tal como está definido.
Prueba de Cubrimiento	Es el acto de asegurar que toda línea de código se ejercita al menos una vez.
Revisión de Diseño	Una revisión técnica en la cual se inspecciona un modelo de diseño.
Prueba de Regresión de Herencia	Es el acto de ejecutar casos de prueba de las súper clases, tanto de forma directa como indirecta en una subclase específica.
Prueba de Integración	Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntas.
Prueba de Método	Consiste en realizar pruebas para verificar que un método (función miembro) funciona tal como está definido.
Revisión de Modelos	Un tipo de inspección, que puede ser desde una revisión técnica formal hasta un recorrido informal, realizado por personas diferentes a las que estuvieron directamente involucradas en el desarrollo del modelo.
Prueba de Caminos	Es el acto de asegurar que todos los caminos lógicos en el código se ejercitan al menos una vez.
Revisión de Prototipos	Es un proceso mediante el cual los usuarios trabajan a través de una colección de casos de uso, utilizando un prototipo como si fuera el sistema real. El objetivo principal es probar si el diseño

	del prototipo satisface las necesidades de esos usuarios.
Demostrar con el código	La mejor forma de determinar si un modelo realmente refleja lo que se necesita, o lo que se debe construir, es construyendo software basado en el modelo para mostrar que el modelo está bien
Prueba de Regresión	El acto de asegurar que los comportamientos previamente probados todavía trabajan como se espera luego que se han realizado cambios a la aplicación.
Prueba de Stress	El acto de asegurar que el sistema funciona como se espera bajo grandes volúmenes de transacciones, usuarios, carga y demás.
Revisión Técnica	Una técnica de aseguramiento de la calidad en la cual el diseño de la aplicación es revisado de forma exhaustiva por un grupo encargado. Una revisión típicamente se enfoca en la precisión, calidad, facilidad de uso y completitud. A este proceso usualmente se le llama recorrido, inspección, o revisión de encargados.
Prueba de Escenarios de Uso	Una técnica de prueba en la cual una o más personas validan un modelo siguiendo la lógica de los escenarios de uso.
Prueba de Interfaz de Usuario	Consiste en probar la interfaz de usuario para garantizar que cumple los estándares y requerimientos definidos. Usualmente se refiere a la prueba de interfaz de usuario gráfico.
Prueba de Caja-Blanca	Consiste en realizar pruebas para verificar que líneas específicas de código funcionan tal como está definido. También se le conoce como prueba de caja-transparente.

Fuente: Tomada de <http://www.ambysoft.com/essays/flootSpanish.html>

Descripción del ambiente de pruebas

A continuación se describe la construcción del ambiente de pruebas para la aplicación web:

- Luego de concluir el desarrollo de la aplicación web se procederá a instalar el motor de base de datos MySQL versión 5.1 en el ambiente de pruebas y se ejecutarán los scripts con las sentencias SQL (Estándar Query Language) de creación de base de datos, tablas y registros. A continuación se instalará el servidor de aplicaciones GlassFish versión 3 y por medio de su página de administración se configurará la conexión a la base de datos y el pool de conexiones que será utilizado por la aplicación web.

Al compilar el proyecto, el entorno de desarrollo integrado Netbeans generará el archivo .ear (Enterprise Archive) que contiene el conjunto de clases, librerías y archivos de configuración que hacen parte de la aplicación web que será desplegada dentro del servidor de aplicaciones por medio de la página de administración que además cuenta con un visualizador de registro de actividades donde se podrá consultar si se presentaron errores durante este procedimiento. De esta forma se puede comprobar que el despliegue fue exitoso ingresando a la dirección <http://localhost:8080/ProjectEA-war/ProjectWebServiceService?wsdl> donde se podrá consultar el archivo de descripción del servicio web y en la dirección <http://localhost:8080/ProjectEA-war/ProjectWebServiceService?Tester> se podrán realizar pruebas sobre los métodos expuestos en el servicio web.

La siguiente descripción corresponde a la generación del ambiente de pruebas para la aplicación móvil:

- Se debe tener en cuenta la configuración de la dirección IP (Internet Protocol) versión 4 que se establece con nombre de dominio propio y no interactúa con una dirección dinámica administrada por el protocolo DHCP en el equipo donde se instale el servidor de aplicaciones, ya que será utilizada en el archivo de configuración del cliente móvil para obtener el archivo de descripción del servicio web (WSDL) y realizar los cambios necesarios en las clases correspondientes.

Al compilar el código fuente del proyecto, el entorno de desarrollo integrado Netbeans generará el archivo .jar (Java Archive) que contiene las clases y librerías correspondientes mientras que el archivo .jad (Java Application Descriptor) contiene la información correspondiente a la aplicación como la ubicación de los íconos y los MIDlets que se utilizan. Dichos archivos serán incorporados a la

memoria interna o extraíble del dispositivo móvil para luego disponer de esta en la instalación del aplicativo.

4. RESULTADOS Y DISCUSIÓN

De acuerdo a las pruebas efectuadas dentro de los estándares tomados en cuenta dentro del deck de pruebas se pueden describir los siguientes resultados:

4.1 MODELO DE NEGOCIO

La carga de los scripts con las sentencias de creación de base de datos, tablas y registros fue exitosa ya que no se presentaron errores durante el proceso. Se realizó la comprobación mediante diferentes consultas que arrojaron registros con la información de la base de datos, de tablas y de usuarios que tendrán acceso a la aplicación móvil. Los scripts que fueron fundamentales para las pruebas descritas en la base de datos se encuentran en el anexo G.

4.2 CASOS DE USO

Las pruebas dedicadas al ingreso del usuario, interacción con el sistema y resultados de las consultas que dependían del rol que tiene cada usuario que ingresa a la aplicación generando conectividad dentro del aplicativo móvil.

Durante el primer ciclo de pruebas en el módulo de funcionalidad se observó un error en el código generado automáticamente por el asistente de creación de un cliente de servicio denominado web móvil que pertenece al entorno de desarrollo NetBeans, el cual fue utilizado para el desarrollo del aplicativo que a través del propio lenguaje de programación se generó el 'debugging' que se entiende como un proceso utilizado para encontrar y disminuir el número de errores al recorrer el código fuente instrucción por instrucción.

Para el segundo ciclo fue resuelto el error detectado en el código generado de acuerdo al error mencionado, lo que permite realizar una verificación

sobre la validación y transformación del resultado de las consultas obteniendo la información requerida por el usuario.

4.3 ARQUITECTURA DEL SISTEMA.

Los siguientes son los aspectos que se tuvieron en cuenta dentro de las pruebas que se realizaron a la arquitectura del sistema:

4.3.1 Conexión

El equipo donde está instalado el servidor de aplicaciones es un equipo portátil marca Toshiba que cumple con las siguientes especificaciones:

- Procesador intel core i5 de 2.53 GHz
- 500 GB de disco duro
- 4 GB de memoria RAM
- Tarjeta de red inalámbrica Broadcom compatible con el estándar IEEE 802.11 b, g y n
- Tarjeta de red de área local (LAN) Atheros 10/100/1000

A este equipo le fue asignada la dirección IP (Internet Protocol) v4 192.168.0.10, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.1 de forma manual y su conexión a la red se realizó de forma cableada a un router D-Link que también ofrece conectividad inalámbrica de 150 mbps.

El dispositivo donde fue instalada la aplicación móvil es un Nokia n95 que cumple con las siguientes especificaciones:

- Procesador de 332 MHz
- 160 MB de almacenamiento.
- 64 MB de memoria RAM

- Conectividad a red inalámbrica compatible con el estándar IEEE 802.11 b y g

Al dispositivo le fue asignada la dirección IP v4 192.168.0.20, la máscara de subred 255.255.255.0 y la puerta de enlace 192.168.0.1 de forma manual para establecer la conexión inalámbrica con el router D-Link y en consecuencia estar en red con el servidor de aplicaciones donde se encuentra desplegado el servicio web. Posteriormente se modificaron las clases pertenecientes a la aplicación móvil a través de un asistente del entorno de desarrollo Netbeans que recibe el siguiente parámetro <http://192.168.0.10:8080/ProjectEA-war/ProjectWebServiceService?WSDL>, el cual contiene la dirección, el puerto y la ruta donde se encuentra el archivo de definición del servicio web (WSDL) para que pueda ser descubierto por la aplicación móvil.

La conexión entre el dispositivo y el servidor de aplicaciones se probó ejecutando la aplicación móvil que muestra en pantalla el formulario donde se solicita el ingreso de las credenciales de un usuario registrado. Se ingresó en el nombre de usuario 'julian.osorio' y en la contraseña '123456' y luego de ejecutar el comando de ingreso se visualizó el menú inicial lo cual indica que la conexión fue exitosa, de lo contrario se hubiera mostrado un error acompañado del mensaje arrojado por cliente del servicio web.

4.3.2 Consultas

La aplicación móvil permite al usuario realizar consultas de bibliografía por título, autor y tema junto a la opción de avanzadas donde se pueden combinar las tres opciones mencionadas. Como resultado se obtiene en pantalla una tabla con las siguientes columnas: ISBN, Título, Autor, Tema, Edición y No. Disponible.

Estas consultas reciben parámetros de tipo alfanumérico debido a que el material perteneciente a la Biblioteca de la Universidad Libre Sede Bosque Popular puede estar conformado por ese patrón y se cuenta con una única restricción donde dichos parámetros pueden estar conformados hasta por veinte (20) caracteres que corresponden a los descriptores del material.

Inicialmente los estudiantes y docentes tienen el rol “usuario”, lo cual indica que pueden realizar el mismo tipo de consultas desde el dispositivo móvil sin ninguna restricción.

4.3.3 Rendimiento

Cuando hablamos del rendimiento de la aplicación hacemos referencia al consumo de memoria disponible en el dispositivo y al tiempo que le toma mostrar los resultados en pantalla.

Con respecto al consumo de memoria del dispositivo se recurre a varias técnicas de optimización del código como el uso de arreglos y no de vectores, no abusar de los bucles o ciclos, obtener resultados a través de servicios web y no de conexiones http que resultan ser costosas en cuestión de recursos, no abusar de las excepciones ya que deben ser lanzadas cuando sea necesario y finalmente liberar recursos e instancias de objetos tan pronto como sea posible.

Se midió el tiempo que le toma a cada consulta mostrar el resultado en la pantalla del dispositivo mientras está conectado a una red inalámbrica con buena y mala intensidad de señal, lo que permitiría obtener mediciones mas aproximadas a un ambiente de producción donde la señal disponible en el campus de la Universidad varía constantemente debido a las condiciones climatológicas, entre otras. Cabe resaltar que no se realizó la prueba con una conexión a la red de datos del operador celular (Movistar), ya que esto

requiere que el servidor de aplicaciones que contiene el servicio web esté publicado en una dirección IP pública para ser consumido.

El tiempo tomado por la aplicación para mostrar el resultado en pantalla luego de ejecutar las diferentes consultas sobre una red inalámbrica cuya intensidad era óptima fue de 2 segundos aproximadamente mientras que el tiempo tomado sobre una red inalámbrica con una intensidad baja fue de 3 segundos aproximadamente.

4.3.4 Protocolos

La conexión entre el servicio web y la aplicación móvil se realiza mediante el protocolo Soap (Simple Object Access Protocol), el cual consiste en el intercambio de mensajes basados en el lenguaje XML (Extensive Markup Language) lo cual deja entrever algunas ventajas:

- Puede ser usado por cualquier lenguaje de programación como Java, php y .Net
- Puede ser asociado a cualquier protocolo que sea capaz de transportar texto como Http.
- Permite la interoperabilidad entre diferentes entornos asegurando la comunicación entre una aplicación de escritorio y una móvil a un servidor a través de Http.

Se observa que el uso del protocolo Soap asegura que la comunicación entre el servicio web y la aplicación móvil sea liviana debido al uso de XML y efectiva porque no será bloqueada por ningún sistema de seguridad debido a que los mensajes intercambiados viajan a través del protocolo Http.

4.3.5 Seguridad

La aplicación móvil ofrece los siguientes niveles de seguridad a los usuarios:

- El ingreso a la aplicación es otorgado luego de un proceso de registro donde se le entregarán al usuario las credenciales compuestas por un nombre de usuario y una contraseña que deberán ser digitadas en el formulario de ingreso que aparece en la pantalla del dispositivo luego de aplicación. Si algún dato ingresado en el formulario no corresponde con dichas credenciales se mostrará un mensaje de advertencia notificando al usuario sobre el error.
- El rol asignado a cada usuario le permite realizar diferentes actividades dentro de la aplicación móvil, por ejemplo, el rol usuario solo permite realizar consultas de bibliografía por título, autor, tema y avanzadas asegurando un control sobre la aplicación.
- La aplicación móvil desarrollada no tiene acceso a la información que se encuentra almacenada en el dispositivo ya que para esto es necesario implementar ciertas librerías que no fueron usadas durante el desarrollo de la misma.

4.3.6 Configuración

La aplicación móvil desarrollada está dirigida a los dispositivos que implementen el perfil MIDP (Mobile Information Device Profile) versión 2.0 y se refiere al nivel de aplicación donde se define un conjunto mínimo de librerías para una determinada serie de dispositivos la cual es proporcionada por la configuración CLDC (Connected Limited Device Configuration) versión 1.0 está diseñada para dispositivos que cuentan con pantallas pequeñas, poco poder de procesamiento y memoria, además implementa la máquina virtual KVM (Kilo Virtual Machine). Con lo mencionado anteriormente se

asegura la compatibilidad con un amplio número de dispositivos que no son considerados “inteligentes” porque apenas cuentan con un total de 2 MB de memoria.

4.4 NAVEGACIÓN DEL SISTEMA

La navegación del sistema se puede consultar en el anexo E que corresponde al manual de usuario.

5. CONCLUSIONES

- La conexión entre dispositivo móvil y el servicio web se configura fácilmente mediante un asistente proporcionado por el entorno de desarrollo integrado Netbeans. Si hay un cambio en la dirección IP del servidor de aplicaciones donde se encuentre alojada la aplicación web, el asistente requerirá la nueva dirección junto con la ruta del archivo de descripción del servicio web (WSDL).
- El usuario podrá generar consultas por título, autor, tema y avanzada, la cual permite combinar los tipos mencionados anteriormente para obtener resultados más completos.
- La aplicación móvil no tendrá acceso a ningún tipo de información que se encuentra almacenada en el dispositivo donde sea instalada y en caso de su pérdida o hurto no se podrá acceder a la aplicación ya que será necesario ingresar las credenciales suministradas con cada uso de la aplicación.
- Los protocolos Soap y Http permiten que los tiempos de respuesta de la aplicación móvil sean menores debido a la transferencia de mensajes Xml comparado con el uso exclusivo de Http que requiere más recursos.
- Con la implementación del deck el proceso de generación de pruebas es más organizado, permitiendo enfocar el esfuerzo en los distintos aspectos que componen todo el software como son: el rendimiento, la configuración, la lógica del negocio y funcionalidad.
- La implementación de la configuración CLDC versión 1.0 y el perfil MIDP versión 2.0 asegura la compatibilidad de la aplicación móvil desarrollada con un amplio número de dispositivos que se encuentran disponibles en el mercado.

- La aplicación móvil le ofrece al usuario una navegación sencilla que se puede manejar por medio del teclado del dispositivo y se hará énfasis de la misma en el anexo E que corresponde al manual del usuario.
- Para incentivar el uso de la aplicación móvil de consulta de bibliografía entre la comunidad académica, se propone realizar un evento que será difundido a través del correo institucional y avisos en las carteleras de las diferentes facultades, donde los participantes se pueden acercar con su dispositivo para realizar pruebas de instalación y funcionamiento.

6. RECOMENDACIONES

- La arquitectura de desarrollo del software es escalable, lo que permite adaptar nuevas funcionalidades dentro de la aplicación, para lo cual se recomienda dentro de una segunda fase la posibilidad de realizar solicitudes de reserva y préstamo de material bibliográfico.
- La tecnología Java ME que fue utilizada para desarrollar la aplicación móvil cubre un número limitado de dispositivos que dependen del fabricante y de las últimas tecnologías de generación disponibles en el mercado a lo cual se recomienda portar la aplicación a otros sistemas operativos como iOS de Apple y Android de Google.
- El objeto de desarrollo del aplicativo móvil permitirá generar una cultura con una nueva herramienta para la consulta de bibliografía de manera virtual y para esto se recomienda implementar nuevas funcionalidades que proyecten su alcance.
- El desarrollo de aplicaciones móviles permitirá compartir materiales de interés dentro de la comunidad académica perteneciente a la Universidad Libre sede Bosque Popular.

BIBLIOGRAFÍA

FROUFE, Agustín. y JORGE, P. (2006). *J2ME Java 2 Micro Edition Manual de usuario y tutorial*. Bogotá: Alfaomega Colombiana S.A.

KROLL, P. y KRUCHTEN, P. (2003). *Rational Unified Process Made Easy: A Practitioner's Guide to the RUP, The*. Addison Wesley.

RAMÍREZ, P. L. (2006). *visualizagr.jsp*. Recuperado el 20 de 5 de 2010, de 201.234.78.173:8080/gruplac:
<http://201.234.78.173:8080/gruplac/jsp/visualiza/visualizagr.jsp?nro=000000000004448>

VERA, J. e YBARRA, G. B. (2011). *kohaj2me.pl*. Recuperado el 20 de 5 de 2010, de biblio.ing.unlp.edu.ar: <http://biblio.ing.unlp.edu.ar/cgi-bin/koha/kohaj2me.pl>

INFOGRAFÍA

- <http://libraries.mit.edu/>
- <http://www.ox.ac.uk/research/libraries/index.html>
- [http://solo.bodleian.ox.ac.uk/primo_library/libweb/action/search.do?ct=Next+Page&pag=nxt&dscnt=0&frbg=&scp.scps=scope%3A\(%22OX%22\)&tab=local&dstmp=1302307187365&srt=rank&vl\(217121274UI0\)=aany&vl\(243207771UI1\)=all_items&ct=Next%20Page&mode=Basic&dum=true&indx=31&tb=t&vl\(1UIStartWith0\)=contains&vl\(freeText0\)=j2ee&vid=OXVU1&fn=search](http://solo.bodleian.ox.ac.uk/primo_library/libweb/action/search.do?ct=Next+Page&pag=nxt&dscnt=0&frbg=&scp.scps=scope%3A(%22OX%22)&tab=local&dstmp=1302307187365&srt=rank&vl(217121274UI0)=aany&vl(243207771UI1)=all_items&ct=Next%20Page&mode=Basic&dum=true&indx=31&tb=t&vl(1UIStartWith0)=contains&vl(freeText0)=j2ee&vid=OXVU1&fn=search)
- http://solo.bodleian.ox.ac.uk/primo_library/libweb/action/search.do?menuitem=0&vid=OXVU1&fromTop=true&fromPreferences=false&fromEsself=false&vid=OXVU1
- <http://biblos.javeriana.edu.co/uhtbin/webcat>
- <http://bdigital.eafit.edu.co:8080/sinbad/>
- <http://catalogo.urosario.edu.co/uhtbin/cgisirsi/kA7JxvfoW9/BIBARA/198770022/60/502/X>
- http://www.catalogo.unal.edu.co/F/-/?func=find-b-0&local_base=UNC01
- <http://www.devx.com/wireless/Article/28046/1954>
- http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=%2Fcom.ibm.websphere.wsfepl.multiplatform.doc%2Finfo%2Fae%2Fae%2Ftwbs_jaxbjava2schema.html

ANEXO A

Entrevista realizada a Liliana López, directora de la Biblioteca de la Universidad Libre Sede Bosque Popular

¿Cuáles son las características del Sistema SIABUC?

- No se pueden hacer consultas entre seccionales – se piensa hacer a futuro
- Arroja estadísticas de libros más/menos consultados
- Se utiliza el Sistema de Clasificación Decimal Dewey para la clasificación de los títulos
- Las revistas solo pueden ser prestadas para consulta interna
- No hay mapas pero se pueden consultar atlas
- Para las tesis hay un manejo especial
- Los descriptores son palabras clave para consultar un libro cuando el usuario no conoce el título/autor

¿Qué tipo de tesauros son manejados?

Los tesauros son especializados en algunas materias y manejan nomenclaturas:

- TR: términos relacionados
- TE: términos específicos

¿Cómo es el proceso de préstamo de libros?

- Se realiza una consulta en la base de datos para ubicar el libro.

- El número máximo de libros por préstamo es de tres por ocho días.
- El sistema cuenta con la opción de préstamos.
- Se debe ingresar el código del estudiante y a continuación se muestran los libros vencidos y prestados.
- Se registra el código del libro y se obtiene su información.
- Para la devolución del libro se requiere únicamente el código del mismo.

¿Existen estadísticas de consultas realizadas por estudiantes?

- Solo se cuenta con estadísticas por facultades y libros más consultados.

¿Se cuenta con estadísticas de consumo de bases de datos externas como Proquest?

- Se cuenta con estadísticas proquest 2008 y 2009.

¿Cómo se realiza el plan de compras de libros?

- Se envía por correo electrónico un formato de solicitud de libros a los decanos. Luego este formato se difunde a los directores de programa.
- Los directores de programa se reúnen con los docentes para diligenciar el formato.
- Hasta el día quince (15) de cada mes se reciben los formatos diligenciados.

¿Cuál es la relación de búsqueda de libros?

- Se seleccionan las palabras claves del contenido cuando no aplica el “LEM”

¿Cómo funcionan los préstamos interbibliotecarios?

- Se envía un correo a principio de semestre a las bibliotecas de las otras universidades adjuntando una carga para reiterar el convenio
- Para los préstamos interbibliotecarios hay 2 formatos: consulta y préstamo.
- Se requieren los datos del libro: autor, título, # clasificación
- Se entrega una carta por 10 días.

¿Existe un formato para el préstamo de libros?

- Si el sistema se cae se utiliza un formato para el préstamo.

ANEXO B
Formato de la encuesta realizada a estudiantes y docentes de la
Universidad Libre Sede Bosque Popular

UNIVERSIDAD LIBRE
FACULTAD DE INGENIERIA
INGENIERIA DE SISTEMAS E INFORMATICA

ESTUDIANTE: _____ DOCENTE: _____
PROGRAMA: _____
SEMESTRE: _____

1. ¿Conoce el Sistema de Información actual para consulta de Bibliografía?
a) Si
b) No
 Ⓢ ¿Por qué?
2. ¿Realiza consultas frecuentes a través del Sistema de Información de la Biblioteca?
a) Si
b) No
 Ⓢ ¿Por qué?
3. ¿Cree que el Sistema de Información de la Biblioteca es apropiado?
a) Si
b) No
 Ⓢ ¿Por qué?
4. ¿Realiza consultas a Bases de Datos como Proquest?
a) Si
b) No
 Ⓢ ¿Por qué?
5. ¿Realiza consultas a Catálogos de otras Bibliotecas y Universidades?
a) Si
b) No
 Ⓢ ¿Por qué?

ANEXO C			
Resultados de la encuesta realizada a estudiantes y docentes de la Universidad Libre Sede Bosque Popular			
Estudiantes	45		95.74468085
Docentes	2		4.255319149
Total Encuestados	47		

1.	Si	20		42.55319149
	No	27		57.44680851

2.	Si	18		38.29787234
	No	29		61.70212766

3.	Si	27		57.44680851
	No	15		31.91489362
	NR	5		10.63829787

4.	Si	7		14.89361702
	No	40		85.10638298

5.	Si	17		36.17021277
	No	30		63.82978723

6.	Si	44		93.61702128
	No	3		6.382978723

	Ed. Física	Ing. Ambiental	Ing. Mecánica	Ing. Sistemas
Estudiantes	18	10	2	9
Docentes	0	1	0	0

	Ing. Industrial	Derecho	Ped. Infantil	Idiomas
Estudiantes	1	3	2	0
Docentes	0	0	0	1

ANEXO D

Código fuente desarrollado para el proyecto

```
package org.app.project.midlets;

import javax.microedition.lcdui.Alert;

import javax.microedition.lcdui.AlertType;

import javax.microedition.lcdui.Choice;

import javax.microedition.lcdui.Command;

import javax.microedition.lcdui.CommandListener;

import javax.microedition.lcdui.Display;

import javax.microedition.lcdui.Displayable;

import javax.microedition.lcdui.Form;

import javax.microedition.lcdui.List;

import javax.microedition.lcdui.TextField;

import javax.microedition.midlet.*;

import org.app.project.forms.ConsultaAvanzadaForm;

import org.app.project.forms.LoginForm;

import org.app.project.forms.ResultadoForm;
```

```

import org.app.project.forms.TestForm;

import org.app.project.wsclient.ProjectWebServiceService_Stub;

import org.app.project.wsclient.libro;

import org.netbeans.microedition.lcdui.SimpleTableModel;

import org.netbeans.microedition.lcdui.TableItem;


public class TestMidlet extends MIDlet implements CommandListener,
Runnable {

    private Display display;

    private TestForm form;

    private LoginForm loginForm;

    private ConsultaAvanzadaForm formConsultaAvanzada;

    private ResultadoForm formResultado;

    private List listaMenu;

    private List testLista;

    private String query = null;

    private String query1 = null;

    private String query2 = null;

```

```
private String query3 = null;

private String label = null;

private boolean[] elementosSeleccionados;

private ProjectWebServiceService_Stub service;

private TableItem tableItem;

private libro[] lib;

private Thread proceso;

private boolean completo = false;


public void startApp() {

    display = Display.getDisplay(this);

    loginForm = new LoginForm("Ingreso");

    loginForm.setCommandListener(this);

    display.setCurrent(loginForm);

}
```

```

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

/**
 * metodo que retorna el menu que se muestra al usuario
 * estudiante/docente
 * @return
 */

public void menuUsuario() {

    listaMenu = new List("Menu", Choice.IMPLICIT);

    listaMenu.append("Consulta por Titulo", null);

    listaMenu.append("Consulta por Autor", null);

    listaMenu.append("Consulta por Tema", null);

    listaMenu.append("Consulta Avanzada", null);

    listaMenu.setCommandListener(this);

```

```

        display.setCurrent(listaMenu);

    }

    /**
     * metodo para validar usuario logueado
     */
    private void validarUsuario() {

        iniciaProceso();

    }

    /**
     * metodo que instancia un objeto de TestForm y lo muestra en pantalla
     */
    private void formTitulo() {

        String cadena = "titulo";

        form = new TestForm("Titulo", cadena);

        form.setCommandListener(this);

        display.setCurrent(form);
    }

```

```
}
```

```
/**
```

```
 * metodo que instancia un objeto de TestForm y lo muestra en pantalla
```

```
 */
```

```
private void formAutor() {
```

```
    String cadena = "autor";
```

```
    form = new TestForm("Autor", cadena);
```

```
    form.setCommandListener(this);
```

```
    display.setCurrent(form);
```

```
}
```

```
/**
```

```
 * metodo que instancia un objeto de TestForm y lo muestra en pantalla
```

```
 */
```

```
private void formTema() {
```

```
    String cadena = "tema";
```

```
    form = new TestForm("Tema", cadena);
```

```
    form.setCommandListener(this);
```

```

        display.setCurrent(form);

    }

    /**
     * metodo encargado de mostrar en pantalla el formulario de consulta de
     * bibliografia avanzada
     */

    private void testConsultaAvanzada() {

        testLista = new List("Consulta Avanzada", Choice.MULTIPLE);

        Command comandoConsultar = new Command("Consultar",
        Command.OK, 1);

        Command comandoAtras = new Command("Atras", Command.BACK,
        0);

        Command comandoSalir = new Command("Salir", Command.EXIT, 2);

        testLista.append("Consulta por Titulo", null);

        testLista.append("Consulta por Autor", null);

        testLista.append("Consulta por Tema", null);

        testLista.addCommand(comandoConsultar);

        testLista.addCommand(comandoAtras);

        testLista.addCommand(comandoSalir);
    }

```



```

        testLista.setCommandListener(this);

        display.setCurrent(testLista);
    }

    /**
     * metodo para iniciar el hilo
     */
    private void iniciaProceso() {

        System.out.println("inicia proceso");

        proceso = new Thread(this);

        proceso.start();

        System.out.println("fin de proceso");

    }

    /**
     * metodo donde se transforma un arreglo de tipo libro a un arreglo de
     arreglos de tipo String
     * @param libro[] lib

```

```

* @return String[][]

*/

private String[][] arrayToArrayOfArrays(libro[] lib) {

    int posiciones = lib.length;

    int elementos = 6;

    String[][] tableData = new String[posiciones][elementos];

    for (int i = 0; i < lib.length; i++) {

        libro libro = new libro();

        libro = lib[i];

        tableData[i][0] = libro.getIsbn();

        tableData[i][1] = libro.getTitulo();

        tableData[i][2] = libro.getAutor();

        tableData[i][3] = libro.getTema();

        tableData[i][4] = libro.getEdicion();

        tableData[i][5] = Integer.toString(libro.getDisponible());

    }

    return tableData;

```

```

    }

    /**

    * metodo que retorna un objeto TableItem con el resultado de la consulta
    para mostrar al usuario

    * @param tableData

    * @return TableItem

    */

    private TableItem retornaTableItem(String[][] tableData) {

        String[] tableHeaders = new String[]{"Isbn", "Titulo", "Autor", "Tema",
        "Edicion", "No. Disponible"};

        SimpleTableModel model = new SimpleTableModel(tableData,
        tableHeaders);

        tableItem = new TableItem(display, "Resultado", model);

        return tableItem;

    }

    /**

    * metodo que muestra un mensaje de alerta en pantalla

```

```

* @param titulo

* @param mensaje

*/

private void muestraAlerta(String titulo, String mensaje) {

    try {

        Alert alerta = new Alert(titulo, mensaje, null, AlertType.ERROR);

        alerta.setTimeout(Alert.FOREVER);

        display.setCurrent(alerta, listaMenu);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

/**

* metodo que muestra un mensaje de alerta en pantalla

* @param titulo

* @param mensaje

* @param e

*/

```

```

private void muestraAlerta(String titulo, String mensaje, Exception e) {

    try {

        Alert alerta = new Alert(titulo, mensaje + e.getMessage(), null,
AlertType.ERROR);

        alerta.setTimeout(Alert.FOREVER);

        display.setCurrent(alerta, listaMenu);

    } catch (Exception ex) {

        ex.printStackTrace();

    }

}

```

```

/**

```

```

 * metodo que muestra un mensaje de alerta en pantalla

```

```

 * @param titulo

```

```

 * @param mensaje

```

```

 * @param formulario

```

```

 */

```

```

private void muestraAlerta(String titulo, String mensaje, Form formulario) {

    try {

```

```

        Alert alerta = new Alert(titulo, mensaje, null, AlertType.ERROR);

        alerta.setTimeout(Alert.FOREVER);

        display.setCurrent(alerta, formulario);

    } catch (Exception e) {

        e.printStackTrace();

    }

}

/**
 * metodo que muestra un mensaje de alerta en pantalla
 *
 * @param titulo
 *
 * @param mensaje
 *
 * @param e
 *
 * @param formulario
 */

private void muestraAlerta(String titulo, String mensaje, Exception e, Form
formulario) {

    try {

        Alert alerta = new Alert(titulo, mensaje + e.getMessage(), null,
AlertType.ERROR);

```

```

        alerta.setTimeout(Alert.FOREVER);

        display.setCurrent(alerta, formulario);

    } catch (Exception ee) {

        ee.printStackTrace();

    }

}

/**

 * metodo implementado por la interface Runnable donde se realiza el
 llamado al servicio web

 */

public void run() {

    try {

        System.out.println("Label " + label);

        if ("Ingresar".equals(label)) {

            System.out.println("Llamado al WS de validacion");

            service = new ProjectWebServiceService_Stub();

            try {

```

```

        System.out.println(service.testMethod());

        boolean test =
service.userLogin(loginForm.getCampoUsuario().getString(),
loginForm.getCampoClave().getString());

        if (test) {

            System.out.println("Llamado al WS de rol");

            int rol =
service.obtenerRolUsuario(loginForm.getCampoUsuario().getString());

            if (rol == 0) {

                muestraAlerta("Error", "El usuario no tiene un rol asignado",
loginForm);

            } else if (rol == 1) {

                menuUsuario();

            }

            } else {

                muestraAlerta("Login", "Los datos ingresados son
incorrectos", loginForm);

            }

        } catch (Exception e) {

            muestraAlerta("Error", "Error", e, loginForm);

        }

```



```

    } else if ("Consultar Titulo".equals(label)) {

        System.out.println("Implementar el llamado al WS de titulo");

        service = new ProjectWebServiceService_Stub();

        lib = null;

        try {

            System.out.println(service.testMethod());

            lib = service.consultaLibroPorTitulo(query);

            System.out.println("lib " + lib.length);

        } catch (Exception e) {

            System.out.println("Exception " + e.getMessage());

            muestraAlerta("Importante", "Se ha presentado un error", e,
form);

        }

        if (lib.length == 0) {

            muestraAlerta("Importante", "No hay resultados", form);

        } else {

```

```

        String[][] tableData = arrayToArrayOfArrays(lib);

        tableItem = retornaTableItem(tableData);

        formResultado = new ResultadoForm("Resultado Consulta");

        formResultado.append(tableItem);

        formResultado.setCommandListener(this);

        display.setCurrent(formResultado);

    }

} else if ("Consultar Autor".equals(label)) {

    System.out.println("Implementar el llamado al WS de autor");


    service = new ProjectWebServiceService_Stub();

    lib = null;

    try {

        System.out.println(service.testMethod());

        lib = service.consultaLibroPorAutor(query);

        System.out.println("lib " + lib.length);

    } catch (Exception e) {

        System.out.println("Exception " + e.getMessage());
    }
}

```

```
muestraAlerta("Importante", "Se ha presentado un problema ", e,  
form);
```

```
}
```

```
if (lib.length == 0) {
```

```
    muestraAlerta("Importante", "No hay resultados", form);
```

```
} else {
```

```
    String[][] tableData = arrayToArrayOfArrays(lib);
```

```
    tableItem = retornaTableItem(tableData);
```

```
    formResultado = new ResultadoForm("Resultado Consulta");
```

```
    formResultado.append(tableItem);
```

```
    formResultado.setCommandListener(this);
```

```
    display.setCurrent(formResultado);
```

```
}
```

```
} else if ("Consultar Tema".equals(label)) {
```

```
    System.out.println("Implementar el llamado al WS de tema");
```

```
    service = new ProjectWebServiceService_Stub();
```

```
    lib = null;
```

```

try {

    System.out.println(service.testMethod());

    lib = service.consultaLibroPorTema(query);

    System.out.println("lib " + lib.length);

} catch (Exception e) {

    System.out.println("Exception " + e.getMessage());

    muestraAlerta("Importante", "Se ha presentado un problema ", e,
form);

}

if (lib.length == 0) {

    muestraAlerta("Importante", "No hay resultados", form);

} else {

    String[][] tableData = arrayToArrayOfArrays(lib);

    tableItem = retornaTableItem(tableData);

    //form.append(tableItem);

    formResultado = new ResultadoForm("Resultado Consulta");

    formResultado.append(tableItem);

    formResultado.setCommandListener(this);

```

```

        display.setCurrent(formResultado);

    }

    } else if ("Buscar".equals(label)) {

        System.out.println("Implementar el llamado al WS de Consulta
Avanzada");

        service = new ProjectWebServiceService_Stub();

        lib = null;

        if (query1 != null && query2 != null && query3 != null) {

            try {

                System.out.println(service.testMethod());

                lib = service.consultaTituloAutorTema(query1, query2,
query3);

                System.out.println("lib " + lib.length);

            } catch (Exception e) {

                System.out.println("Exception " + e);

                muestraAlerta("Importante", "Se ha presentado un probelma",
e, formConsultaAvanzada);

            }

            if (lib.length == 0) {

```

```

        muestraAlerta("Importante", "No hay resultados",
formConsultaAvanzada);

    } else {

        String[][] tableData = arrayToArrayOfArrays(lib);

        tableItem = retornaTableItem(tableData);

        //form.append(tableItem);

        formResultado = new ResultadoForm("Resultado Consulta");

        formResultado.append(tableItem);

        formResultado.setCommandListener(this);

        display.setCurrent(formResultado);

    }

} else if (query1 != null && query2 != null) {

    try {

        System.out.println(service.testMethod());

        lib = service.consultaTituloAutor(query1, query2);

        System.out.println("lib " + lib.length);

    } catch (Exception e) {

        System.out.println("Exception " + e.getMessage());

        muestraAlerta("Importante", "Se ha presentado un problema",
e, formConsultaAvanzada);

```

```

    }

    if (lib.length == 0) {

        muestraAlerta("Importante", "No hay resultados",
formConsultaAvanzada);

    } else {

        String[][] tableData = arrayToArrayOfArrays(lib);

        tableItem = retornaTableItem(tableData);

        formResultado = new ResultadoForm("Resultado Consulta");

        formResultado.append(tableItem);

        formResultado.setCommandListener(this);

        display.setCurrent(formResultado);

    }

} else if (query1 != null && query3 != null) {

    try {

        System.out.println(service.testMethod());

        lib = service.consultaTituloTema(query1, query3);

        System.out.println("lib " + lib.length);
    }
}

```

```

    } catch (Exception e) {

        System.out.println("Exception " + e);

        muestraAlerta("Importante", "Se ha presentado un problema",
e, formConsultaAvanzada);

    }

    if (lib.length == 0) {

        muestraAlerta("Importante", "No hay resultados",
formConsultaAvanzada);

    } else {

        String[][] tableData = arrayToArrayOfArrays(lib);

        tableItem = retornaTableItem(tableData);

        formResultado = new ResultadoForm("Resultado Consulta");

        formResultado.append(tableItem);

        formResultado.setCommandListener(this);

        display.setCurrent(formResultado);

    }

} else if (query2 != null && query3 != null) {

    try {

        System.out.println(service.testMethod());

```



```

        lib = service.consultaAutorTema(query2, query3);

        System.out.println("lib " + lib.length);

    } catch (Exception e) {

        System.out.println(("Exception " + e));

        muestraAlerta("Importante", "Se ha presentado un problema",
e, formConsultaAvanzada);

    }

    if (lib.length == 0) {

        muestraAlerta("Importante", "No hay resultados",
formConsultaAvanzada);

    } else {

        String[][] tableData = arrayOfArrays(lib);

        tableItem = retornaTableItem(tableData);

        formResultado = new ResultadoForm("Resultado Consulta");

        formResultado.append(tableItem);

        formResultado.setCommandListener(this);

        display.setCurrent(formResultado);

    }

```

```

    }

    formResultado = null;

    query1 = null;

    query2 = null;

    query3 = null;

    }

    } catch (Exception e) {

        muestraAlerta("Error", "Se ha presentado un problema", e);

    }

}

```

```

/**

```

* metodo implementado por la interface CommandListener donde se ejecutan las acciones relacionadas con los menu

```

* @param c

```

```

* @param d

```

```

*/

```

```

public void commandAction(Command c, Displayable d) {

```

```

label = c.getLabel();

System.out.println("label " + label);


if (label.equals("Ingresar")) {

    validarUsuario();

} else if (label.equals("Salir")) {

    destroyApp(true);

    notifyDestroyed();

} else if (label.equals("Atras")) {

    display.setCurrent(listaMenu);

} else if (label.equals("Consultar Titulo")) {

    System.out.println("Funciona Titulo");

    query = form.getTextField().getString();

    if (query == null || query.equals("")) {

        muestraAlerta("Importante", "Debe ingresar el Titulo del libro");

    } else {

        System.out.println("Query Titulo " + query);

        System.out.println("Inicia el Thread");

        iniciaProceso();
    }
}

```

```
}
```

```
} else if (label.equals("Consultar Autor")) {
```

```
    System.out.println("Funciona Autor");
```

```
    query = form.getTextField().getString();
```

```
    if (query == null || query.equals("")) {
```

```
        muestraAlerta("Importante", "Debe ingresar el Nombre del Autor");
```

```
    } else {
```

```
        System.out.println("Query Autor " + query);
```

```
        System.out.println("Inicia el Thread");
```

```
        iniciaProceso();
```

```
    }
```

```
} else if (label.equals("Consultar Tema")) {
```

```
    System.out.println("Funciona Tema");
```

```
    query = form.getTextField().getString();
```

```
    if (query == null || query.equals("")) {
```

```
        muestraAlerta("Importante", "Debe ingresar el Tema");
```

```
    } else {
```

```

        System.out.println("Query Tema " + query);

        System.out.println("Inicia el Thread");

        iniciaProceso();

    }

} else if (label.equals("Consultar")) {

    System.out.println("Funciona Consulta Multiple");

    elementosSeleccionados = new boolean[testLista.size()];

    System.out.println("Elementos      seleccionados      "      +
testLista.getSelectedFlags(elementosSeleccionados));

    if (testLista.getSelectedFlags(elementosSeleccionados) < 2) {

        muestraAlerta("Importante", "Se deben seleccionar minimo 2
opciones");

    } else {

        query1 = null;

        query2 = null;

        query3 = null;

        formConsultaAvanzada = new ConsultaAvanzadaForm("Consulta
Avanzada");

        for (int i = 0; i < testLista.size(); i++) {

            if (testLista.getString(i).equals("Consulta por Titulo") &&
elementosSeleccionados[i]) {

```

```

        formConsultaAvanzada.setCampoTitulo(new
TextField("Titulo:", "", 20, 0));

formConsultaAvanzada.append(formConsultaAvanzada.getCampoTitulo());

        } else if (testLista.getString(i).equals("Consulta por Autor") &&
elementosSeleccionados[i]) {

        formConsultaAvanzada.setCampoAutor(new
TextField("Autor:", "", 20, 0));

formConsultaAvanzada.append(formConsultaAvanzada.getCampoAutor());

        } else if (testLista.getString(i).equals("Consulta por Tema") &&
elementosSeleccionados[i]) {

        formConsultaAvanzada.setCampoTema(new
TextField("Tema:", "", 20, 0));

formConsultaAvanzada.append(formConsultaAvanzada.getCampoTema());

        }

    }

    formConsultaAvanzada.setCommandListener(this);

    display.setCurrent(formConsultaAvanzada);

}

```

```

    } else if (label.equals("Buscar")) {

        if (formConsultaAvanzada.getCampoTitulo() != null) {

            if (formConsultaAvanzada.getCampoTitulo().getString().equals("")) {

                muestraAlerta("Importante", "Debe ingresar el Titulo del Libro",
formConsultaAvanzada);

            } else {

                query1 = formConsultaAvanzada.getCampoTitulo().getString();

            }

        }

    }

    if (formConsultaAvanzada.getCampoAutor() != null) {

        if (formConsultaAvanzada.getCampoAutor().getString().equals("")) {

            muestraAlerta("Importante", "Debe ingresar el Autor del Libro",
formConsultaAvanzada);

        } else {

            query2 = formConsultaAvanzada.getCampoAutor().getString();

        }

    }

}

```

```

        if (formConsultaAvanzada.getCampoTema() != null) {

            if (formConsultaAvanzada.getCampoTema().getString().equals(""))
            {

                muestraAlerta("Importante", "Debe ingresar el Tema",
formConsultaAvanzada);

            } else {

                query3 = formConsultaAvanzada.getCampoTema().getString();

            }

        }

        iniciaProceso();

    }

    else if(label.equals("Prueba")) {

        int posSelec = tableItem.getSelectedCellRow();

        for(int cont = 0; cont < lib.length; cont ++) {

            if(cont == posSelec) {

                libro libro = new libro();

                libro = lib[cont];

```



```

        System.out.println(libro.toString());

    }

}

}

else {

    List lista = (List) display.getCurrent();

    switch (lista.getSelectedIndex()) {

        case 0:

            formTitulo();

            break;

        case 1:

            formAutor();

            break;

        case 2:

            formTema();

            break;

        case 3:

            testConsultaAvanzada();

    }
}

```

```

    }

}

}

/*

* To change this template, choose Tools | Templates
* and open the template in the editor.

*/

package org.app.project.webservice;


import java.util.logging.Level;

import java.util.logging.Logger;

import javax.ejb.EJB;

import javax.jws.WebMethod;

import javax.jws.WebParam;

import javax.jws.WebService;

import org.app.project.bean.ProjectLocal;

import org.app.project.binding.Libro;

```

```

@WebService()

public class ProjectWebService {

    @EJB

    private ProjectLocal ejbRef;

    // Add business logic below. (Right-click in editor and choose
    // "Web Service > Add Operation"

    @WebMethod(operationName = "testMethod")

    public String testMethod() {

        return ejbRef.testMethod();

    }

    @WebMethod(operationName = "consultaLibroPorTitulo")

    public Libro[] consultaLibroPorTitulo(@WebParam(name = "tituloLibro")
String tituloLibro) throws Exception {

        try {

            return ejbRef.consultaLibroPorTitulo(tituloLibro);

        } catch (Exception e) {

```

```

        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

        throw e;

    }

}

```

```

@WebMethod(operationName = "consultaLibroPorAutor")

public Libro[] consultaLibroPorAutor(@WebParam(name = "autorLibro")
String autorLibro) throws Exception {

    try {

        return ejbRef.consultaLibroPorAutor(autorLibro);

    } catch (Exception e) {

        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

        throw e;

    }

}

```

```

@WebMethod(operationName = "consultaLibroPorTema")

public Libro[] consultaLibroPorTema(@WebParam(name = "temaLibro")
String temaLibro) throws Exception {

```

```

try {

    return ejbRef.consultaLibroPorTema(temaLibro);

} catch (Exception e) {

    Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

    throw e;

}

}

```

```

@WebMethod(operationName = "consultaTituloAutor")

public Libro[] consultaTituloAutor(@WebParam(name = "tituloLibro") String
tituloLibro,

    @WebParam(name = "autorLibro") String autorLibro) throws
Exception {

    try {

        return ejbRef.consultaTituloAutor(tituloLibro, autorLibro);

    } catch (Exception e) {

        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

        throw e;

    }

}

```

```
}
```

```
@WebMethod(operationName = "consultaTituloTema")
```

```
public Libro[] consultaTituloTema(@WebParam(name = "tituloLibro") String  
tituloLibro,
```

```
        @WebParam(name = "temaLibro") String temaLibro) throws  
Exception {
```

```
    try {
```

```
        return ejbRef.consultaTituloTema(tituloLibro, temaLibro);
```

```
    } catch (Exception e) {
```

```
        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,  
"Error " + e.getMessage());
```

```
        throw e;
```

```
    }
```

```
}
```

```
@WebMethod(operationName = "consultaAutorTema")
```

```
public Libro[] consultaAutorTema(@WebParam(name = "autorLibro")  
String autorLibro,
```

```
        @WebParam(name = "temaLibro") String temaLibro) throws  
Exception {
```

```

        try {

            return ejbRef.consultaAutorTema(autorLibro, temaLibro);

        } catch (Exception e) {

            Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

            throw e;

        }

    }

}

@WebMethod(operationName = "consultaTituloAutorTema")

public Libro[] consultaTituloAutorTema(@WebParam(name = "tituloLibro")
String tituloLibro,

        @WebParam(name = "autorLibro") String autorLibro,
        @WebParam(name = "temaLibro") String temaLibro) throws Exception {

    try {

        return ejbRef.consultaTituloAutorTema(tituloLibro, autorLibro,
temaLibro);

    } catch (Exception e) {

        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,
"Error " + e.getMessage());

        throw e;
    }
}

```

```
}  
  
}
```

```
@WebMethod(operationName = "userLogin")  
  
public boolean userLogin(@WebParam(name = "user") String user,  
@WebParam(name = "password") String password) throws Exception {  
  
    try {  
  
        return ejbRef.userLogin(user, password);  
  
    } catch (Exception e) {  
  
        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,  
"Error " + e.getMessage());  
  
        throw e;  
  
    }  
  
}
```

```
@WebMethod(operationName = "obtenerRolUsuario")  
  
public int obtenerRolUsuario(@WebParam(name = "user") String user)  
throws Exception {  
  
    try {  
  
        return ejbRef.obtenerRolUsuario(user);  
  
    }  
  
}
```



```
    } catch (Exception e) {  
  
        Logger.getLogger(this.getClass().getName()).log(Level.SEVERE,  
"Error " + e.getMessage());  
  
        throw e;  
  
    }  
  
}  
  
}
```

ANEXO E

Manual de usuario

1	GENERALIDADES	197
1.1	Presentación	197
1.2	Objetivos	197
2	CARACTERÍSTICA DE OPERACIÓN	197
2.1	Niveles de Acceso.....	197
2.2	Módulos del sistema	198
2.3	Menú principal.....	198
2.3.1	Opciones.....	198
2.3.2	Submenús.....	198
3	OPERACIÓN DEL SISTEMA.....	199
3.1	Explicación de pantallas.....	199
3.2	Operación del programa.....	226
3.3	Interpretación de Reportes.....	227
4	MENSAJES DE ERROR.....	227
4.1	Consideraciones de condiciones de error	227
4.2	Manejo de Errores.....	228
5	GLOSARIO	229

Lista de imágenes

Ilustración 1 - Menú inicial	200
Ilustración 2 - Formulario de Consulta por Título	202
Ilustración 3 – Realizar la consulta por título	203
Ilustración 4 – Mensaje de advertencia cuando no se digita el título	204
Ilustración 5 - La consulta por título no arroja resultados.....	205
Ilustración 6 - Resultado de la consulta por título	206
Ilustración 7 - Formulario de consulta por autor.....	207
Ilustración 8 – Realizar la consulta por autor	208
Ilustración 9 - Mensaje de advertencia cuando no fue digitado el autor	209
Ilustración 10 - La consulta no arroja resultados.....	210
Ilustración 11 - Resultado de la consulta por autor	211
Ilustración 12 - Formulario de consulta por tema	212
Ilustración 13 - Realizar consulta por tema.....	213
Ilustración 14 - Mensaje de advertencia cuando no fue ingresado el tema	214
Ilustración 15 - La consulta por tema no arroja resultados	215
Ilustración 16 - Resultado de la consulta por tema	216
Ilustración 17 - Formulario de consulta avanzada.....	217
Ilustración 18 - Iniciar consulta avanzada	218
Ilustración 19 - Mensaje de advertencia para la consulta avanzada	220
Ilustración 20 - Formulario de consulta avanzada.....	221
Ilustración 21 - Realizar la consulta avanzada.....	222
Ilustración 22 - Mensaje de advertencia cuando no fue digitada información	223
Ilustración 23 – La consulta avanzada no arroja resultados	224
Ilustración 24 - Resultado de la consulta avanzada.....	225

CONTENIDO DEL MANUAL OPERATIVO O MANUAL DE USUARIO

INTRODUCCIÓN

GENERALIDADES

6.1 PRESENTACIÓN

El siguiente documento contiene las características y forma de operación correspondiente a la aplicación móvil para consulta de bibliografía para la biblioteca de la Universidad Libre sede Bosque Popular.

La aplicación móvil fue desarrollada con la tecnología Java J2ME y es compatible con la configuración CLDC 1.0 y perfil MIDP 2.0

6.2 OBJETIVOS

La aplicación móvil para la consulta de bibliografía ofrece los siguientes servicios:

- Permite realizar consultas por título, autor, tema y avanzadas donde el usuario puede seleccionar de una lista los parámetros
- Muestra los resultados de las búsquedas en una forma ordenada y completa

CARACTERÍSTICA DE OPERACIÓN

6.3 NIVELES DE ACCESO

La aplicación no cuenta con un módulo de autenticación de usuarios para su uso. Luego que el usuario ejecute la aplicación, éste ingresará al menú principal donde se encuentran los diferentes tipos de búsqueda

6.4 MÓDULOS DEL SISTEMA

La aplicación móvil para consulta de bibliografía cuenta con los siguientes módulos:

- Módulo de consulta de bibliografía por título
- Módulo de consulta de bibliografía por autor
- Módulo de consulta de bibliografía por tema
- Módulo de consulta avanzada de bibliografía

6.5 MENÚ PRINCIPAL

La aplicación cuenta con un único menú principal

6.5.1 OPCIONES

El menú principal cuenta con las siguientes opciones:

- Consulta por título
- Consulta por autor
- Consulta por tema
- Consulta avanzada

6.5.2 SUBMENÚS

Cada pantalla de la aplicación móvil tendrá un submenú donde aparecerán las diferentes acciones que se pueden realizar.

Los formularios de consulta por título, autor y tema tienen las siguientes opciones en el submenú

- Comando para consultar
- Comando para regresar al menú principal
- Comando para salir de la aplicación

El formulario de consulta avanzada tiene las siguientes opciones en el submenú

- Comando para mostrar los campos de búsqueda basados en la selección de parámetros por parte del usuario
- Comando para regresar al menú principal

- Comando para salir de la aplicación

El formulario que contiene los campos de búsqueda avanzada tiene las siguientes opciones en el submenú

- Comando para consultar
- Comando para regresar al menú principal
- Comando para salir de la aplicación

El formulario que contiene el resultado de la consulta tiene las siguientes opciones en el submenú

- Comando para regresar al menú principal
- Comando para salir de la aplicación

OPERACIÓN DEL SISTEMA

6.6 EXPLICACIÓN DE PANTALLAS

La aplicación móvil para consulta de bibliografía contiene interfaces de usuario sencillas que permiten un uso eficaz



Ilustración 29 - Menú inicial

Esta pantalla muestra un menú con las diferentes consultas que puede realizar el usuario. Éste tiene las siguientes opciones:

- Consulta por título
- Consulta por autor
- Consulta por tema
- Consulta avanzada

Los usuarios pueden recorrerlo con las teclas direccionales hacia arriba y abajo. Para ingresar a los formularios de búsqueda el usuario debe seleccionar la opción del menú y luego presionar la tecla ubicada en el centro de las teclas direccionales. En seguida se mostrará el formulario correspondiente a la selección realizada



Ilustración 30 - Formulario de Consulta por Título

Esta pantalla muestra el formulario donde el usuario puede ingresar el título del libro que quiere consultar. En la parte inferior de la pantalla se muestran los comandos “Atrás” y “Opciones”, que actúan sobre el formulario.

El comando “Atrás” envía al usuario al menú principal, mientras que el comando “Opciones” muestra un submenú con las siguientes opciones:

- Consultar título
- Atrás
- Salir

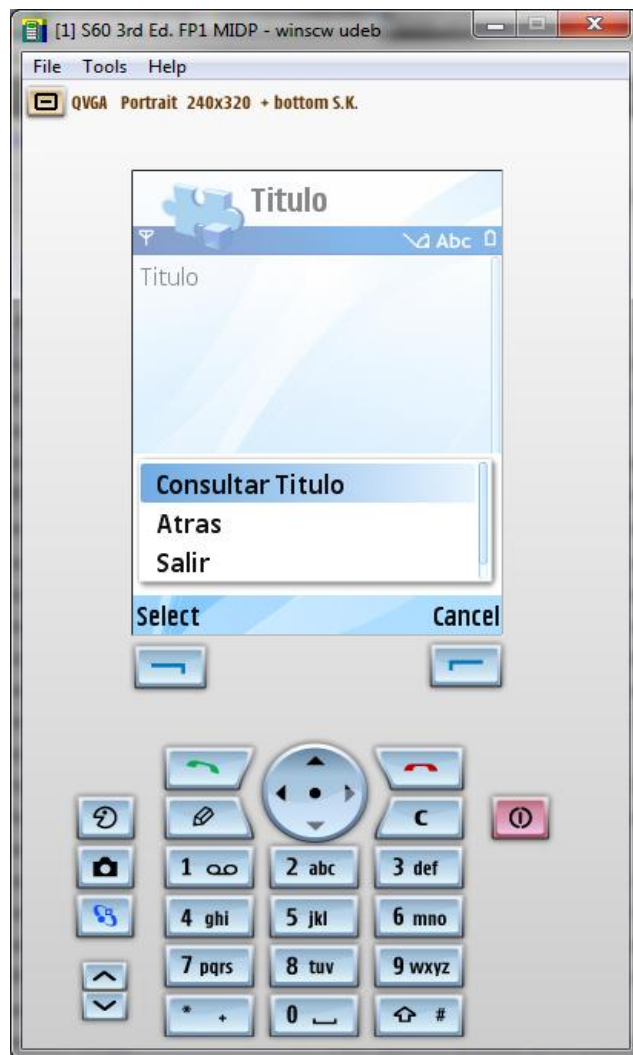


Ilustración 31 – Realizar la consulta por título

Cuando el usuario selecciona la opción “Salir”, la aplicación se cerrará. Cuando el usuario selecciona la opción “Atrás”, la aplicación regresa al formulario.

Cuando el usuario selecciona la opción “Consultar Título” se iniciará el proceso de búsqueda y se validará el formulario.

Si el campo de texto está vacío se mostrará un mensaje de advertencia indicando que se debe digitar el título pero si el usuario ha digitado el título se realiza la consulta con el texto ingresado.



Ilustración 32 – Mensaje de advertencia cuando no se digita el título

Si la consulta no arroja ningún resultado, la aplicación mostrará un mensaje de advertencia

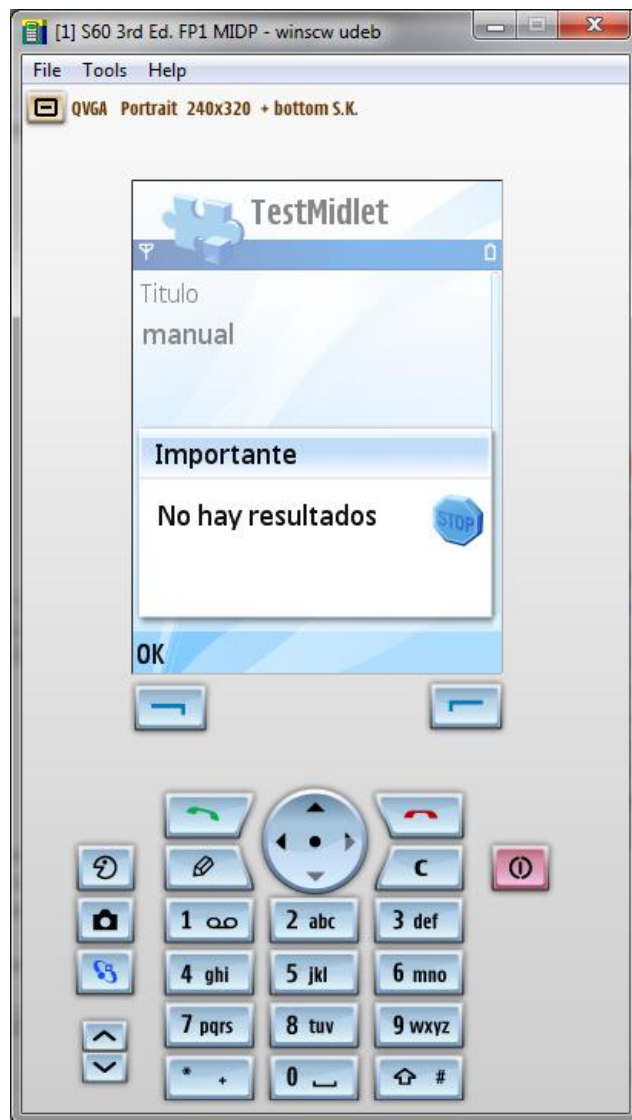


Ilustración 33 - La consulta por título no arroja resultados

Si la consulta arroja resultados, se mostrará al usuario un nuevo formulario que contiene una tabla con las siguientes columnas:

- Isbn
- Título
- Autor
- Tema
- Edición
- No. Disponible

Esta tabla se podrá navegar con el uso del teclado direccional.



Ilustración 34 - Resultado de la consulta por título

En la parte inferior de la pantalla se encuentran los comandos “Atrás” y “Salir” que actúan sobre el formulario. El comando “Atrás” envía al usuario al menú principal y el comando “Salir” cierra la aplicación.



Ilustración 35 - Formulario de consulta por autor

Esta pantalla muestra el formulario donde el usuario puede ingresar el autor del libro que quiere consultar. En la parte inferior de la pantalla se muestran los comandos “Atrás” y “Opciones”, que actúan sobre el formulario.

El comando “Atrás” envía al usuario al menú principal, mientras que el comando “Opciones” muestra un submenú con las siguientes opciones:

- Consultar autor

- Atrás
- Salir



Ilustración 36 – Realizar la consulta por autor

Cuando el usuario selecciona la opción “Salir”, la aplicación se cerrará.

Cuando el usuario selecciona la opción “Atrás”, la aplicación regresa al formulario

Cuando el usuario selecciona la opción “Consultar autor” se iniciará el proceso de búsqueda y se validará el formulario.

Si el campo de texto está vacío se mostrará un mensaje de advertencia indicando que se debe digitar el autor pero si el usuario ha digitado el autor se realiza la consulta con el texto ingresado.



Ilustración 37 - Mensaje de advertencia cuando no fue digitado el autor

Si la consulta no arroja ningún resultado, la aplicación mostrará un mensaje de advertencia

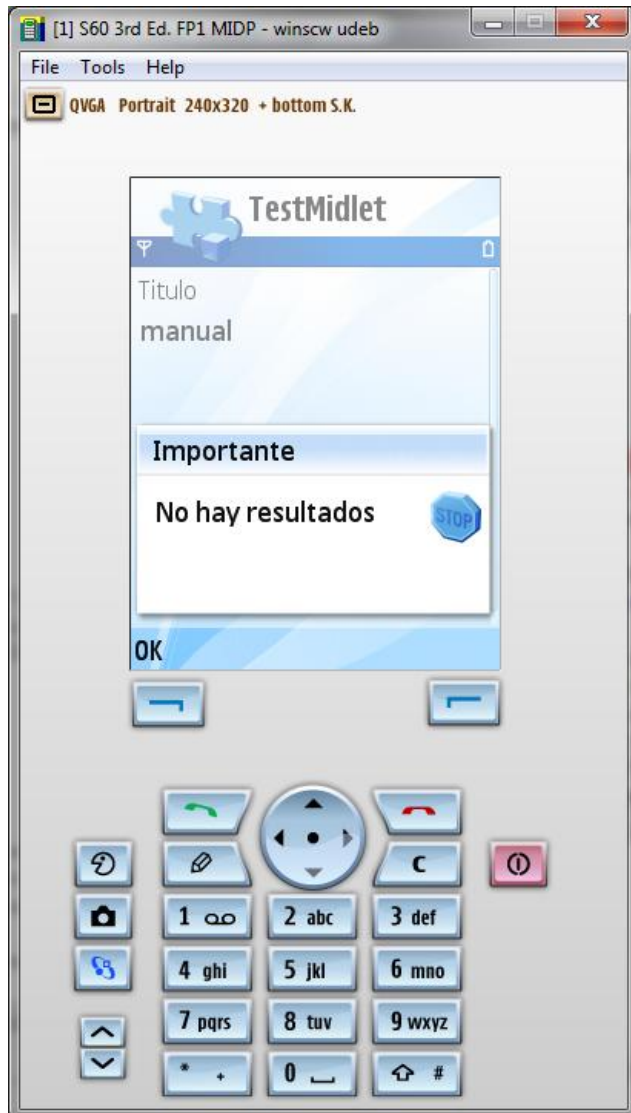


Ilustración 38 - La consulta no arroja resultados

Si la consulta arroja resultados, se mostrará al usuario un nuevo formulario que contiene una tabla con las siguientes columnas:

- Isbn
- Título

- Autor
- Tema
- Edición
- No. Disponible

Esta tabla se podrá navegar con el uso del teclado direccional.

Figura 1.



Ilustración 39 - Resultado de la consulta por autor

En la parte inferior de la pantalla se encuentran los comandos “Atrás” y “Salir” que actúan sobre el formulario. El comando “Atrás” envía al usuario al menú principal y el comando “Salir” cierra la aplicación.



Ilustración 40 - Formulario de consulta por tema

Esta pantalla muestra el formulario donde el usuario puede ingresar el tema del libro que quiere consultar. En la parte inferior de la pantalla se muestran los comandos “Atrás” y “Opciones”, que actúan sobre el formulario.

El comando “Atrás” envía al usuario al menú principal, mientras que el comando “Opciones” muestra un submenú con las siguientes opciones:

- Consultar tema
- Atrás
- Salir



Ilustración 41 - Realizar consulta por tema

Cuando el usuario selecciona la opción “Salir”, la aplicación se cerrará.

Cuando el usuario selecciona la opción “Atrás”, la aplicación regresa al formulario.

Cuando el usuario selecciona la opción “Consultar tema” se iniciará el proceso de búsqueda y se validará el formulario.

Si el campo de texto está vacío se mostrará un mensaje de advertencia indicando que se debe digitar el tema pero si el usuario digitó el tema se realiza la consulta con el texto ingresado



Ilustración 42 - Mensaje de advertencia cuando no fue ingresado el tema

Si la consulta no arroja ningún resultado, la aplicación mostrará un mensaje de advertencia

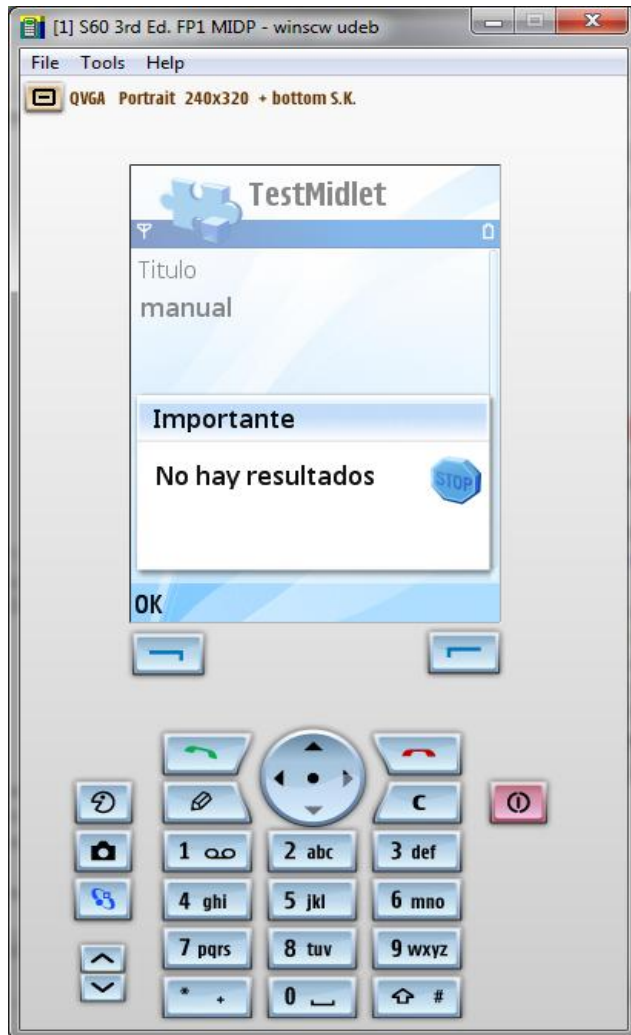


Ilustración 43 - La consulta por tema no arroja resultados

Si la consulta arroja resultados, se mostrará al usuario un nuevo formulario que contiene una tabla con las siguientes columnas:

- Isbn
- Título
- Autor
- Tema
- Edición

- No. Disponible

Esta tabla se podrá navegar con el uso del teclado direccional.



Ilustración 44 - Resultado de la consulta por tema

En la parte inferior de la pantalla se encuentran los comandos “Atrás” y “Salir” que actúan sobre el formulario. El comando “Atrás” envía al usuario al menú principal y el comando “Salir” cierra la aplicación



Ilustración 45 - Formulario de consulta avanzada

Esta pantalla muestra una lista con las opciones de búsqueda: Consulta por título, Consulta por autor y Consulta por tema. Cada opción está acompañada por un checkbox que permite al usuario seleccionar/descartar que parámetros quiere utilizar en la búsqueda.

En la parte inferior de la pantalla se encuentran los comandos “Atrás” y “Opciones”. El comando “Atrás” envía al usuario al menú principal, mientras que el comando “Opciones” contiene un submenú con las siguientes opciones:

- Consultar
- Atrás
- Salir



Ilustración 46 - Iniciar consulta avanzada

El comando “Salir” termina la aplicación, el comando “Atrás” envía al usuario al formulario de Consulta Avanzada y el comando “Consultar” toma los parámetros seleccionados para mostrar a continuación el formulario donde se ingresará la información para la consulta.

Si el usuario no selecciona ninguna opción, la aplicación mostrará un mensaje de advertencia indicando que se deben seleccionar mínimo dos opciones de la lista.

Figura 2.



Ilustración 47 - Mensaje de advertencia para la consulta avanzada

A continuación se muestra un formulario con los campos correspondientes a la selección realizada por el usuario en la lista anterior. En éste, el usuario ingresa la información que quiere consultar.



Ilustración 48 - Formulario de consulta avanzada

En la parte inferior de la pantalla se encuentran los siguientes comandos: “Atrás” y “Opciones”. El comando “Atrás” envía al usuario al menú principal, mientras que el comando “Opciones” tiene un submenú con las siguientes opciones:

- Buscar
- Atrás
- Salir



Ilustración 49 - Realizar la consulta avanzada

El comando “Salir” termina la aplicación, el comando “Atras” envía al usuario al formulario de consulta y el comando “Buscar” inicia el proceso de búsqueda y se validará el formulario.

Si alguno de los campos de texto está vacío, la aplicación muestra un mensaje de advertencia indicando que se deben digitar los parámetros pero si el usuario digitó información se realiza la consulta con el texto ingresado.



Ilustración 50 - Mensaje de advertencia cuando no fue digitada información

Si la consulta no arroja ningún resultado, la aplicación mostrará un mensaje de advertencia.



Ilustración 51 – La consulta avanzada no arroja resultados

Si la consulta arroja resultados, se mostrará al usuario un nuevo formulario que contiene una tabla con las siguientes columnas:

- Isbn
- Título
- Autor
- Tema
- Edición
- No. Disponible

Esta tabla se podrá navegar con el uso del teclado direccional.



Ilustración 52 - Resultado de la consulta avanzada

En la parte inferior de la pantalla se encuentran los comandos “Atrás” y “Salir” que actúan sobre el formulario. El comando “Atrás” envía al usuario al menú principal y el comando “Salir” cierra la aplicación.

6.7 OPERACIÓN DEL PROGRAMA

La operación de la aplicación móvil para la consulta de bibliografía se realiza a través del teclado del dispositivo. Los submenús de cada formulario aparecen sobre los botones conocidos como “soft-keys” y se deben presionar para realizar las diferentes acciones.

Para desplazarse sobre las diferentes opciones del menú principal y de la lista de opciones avanzadas se utiliza las teclas direccionales, en este caso, hacia arriba y abajo. Para ingresar a estas opciones se utiliza la tecla ubicada en el centro de las teclas direccionales y también se utiliza para seleccionar los parámetros a buscar en la consulta avanzada.

En el formulario de consulta avanzada, dependiendo de la selección del usuario, pueden aparecer dos o tres campos de texto donde se debe ingresar la información necesaria. Para esto se utilizan las teclas direccionales hacia arriba y abajo.

El resultado de la consulta muestra una tabla que contiene la información de la bibliografía encontrada basada en el/los parámetro/s ingresado/s por el usuario. Dicha tabla se puede recorrer con las flechas direccionales hacia arriba, abajo, izquierda y derecha.

Los parámetros de consultas se ingresan en los campos de texto utilizando el teclado alfanumérico del dispositivo.

6.8 INTERPRETACIÓN DE REPORTES

El resultado de la consulta será mostrado en una tabla que contiene las siguientes columnas:

- Isbn
- Título
- Autor
- Tema
- Edición
- No. Disponible

La columna ISBN contiene el identificador único de cada libro

La columna Título contiene el título del libro

La columna Autor contiene el autor/es del libro

La columna Tema contiene la temática general del libro

La columna Edición contiene la fecha de edición del libro

La columna No. Disponible contiene el número de ejemplares disponibles.

MENSAJES DE ERROR

6.9 CONSIDERACIONES DE CONDICIONES DE ERROR

- Se presentan mensajes de advertencia indicando al usuario que algún campo de texto en el formulario está vacío luego de ejecutar la acción de consultar.
- Se presenta un mensaje de advertencia indicando que se presenta un error de comunicación al realizar la consulta de bibliografía.
- Se presenta un mensaje de advertencia indicando que la consulta de bibliografía realizada por el usuario no retorna ningún resultado.

6.10 MANEJO DE ERRORES

Se capturan todas las posibles excepciones que se puedan generar durante la ejecución de la aplicación móvil para consulta de bibliografía para que no se vea interrumpida.

GLOSARIO

HTTP: Protocolo de transporte ampliamente conocido. Se basa en el paso de texto mediante redes IP.

SOAP: Protocolo basado en XML que manejan los Web Services para su comunicación.

WebService: Aplicación Web capaz de ofrecer datos o servicios de procesamiento independientemente del lenguaje y la plataforma utilizada.

WSDL: Lenguaje de descripción del Web Service. Detalla los servicios en línea que ofrece un Web Service en particular.

ANEXO F

Se adjunta la hoja de cálculo

ANEXO G

Scripts de base de datos

```
CREATE DATABASE IF NOT EXISTS `test` /*!40100 DEFAULT
CHARACTER SET latin1 */;
```

```
USE `test`;
```

```
-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)
```

```
--
```

```
-- Host: localhost Database: test
```

```
-- -----
```

```
-- Server version 5.1.50-community
```

```
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
```

```
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
```

```
/*!40101 SET NAMES utf8 */;
```

```
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
```

```
/*!40103 SET TIME_ZONE='+00:00' */;
```

```
/*!40014      SET      @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,  
UNIQUE_CHECKS=0 */;
```

```
/*!40014                                           SET  
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0 */;
```

```
/*!40101      SET      @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
--
```

```
-- Table structure for table `libros`
```

```
--
```

```
DROP TABLE IF EXISTS `libros`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `libros` (  
  `isbn` varchar(200) NOT NULL,  
  `titulo` varchar(100) NOT NULL,  
  `autor` varchar(100) NOT NULL,  
  `tema` varchar(100) NOT NULL,
```



```

`edicion` varchar(100) NOT NULL,

`disponible` int(3) NOT NULL,

PRIMARY KEY (`isbn`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--

-- Dumping data for table `libros`

--

LOCK TABLES `libros` WRITE;

/*!40000 ALTER TABLE `libros` DISABLE KEYS */;

INSERT INTO `libros` VALUES ('0001','Como programar en Java','Deitel &
Deitel','Java','Edicion 6',4),('0002','Acceso a bases de datos con Java-JDBC
2.0','Esteban Nunez, Angel','Java','2',4);

/*!40000 ALTER TABLE `libros` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

```

```
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
*/;
```

```
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

```
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed on 2011-11-21 17:27:37
```

```
CREATE DATABASE IF NOT EXISTS `test` /*!40100 DEFAULT
CHARACTER SET latin1 */;
```

```
USE `test`;
```

```
-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)
```

```
--
```

```
-- Host: localhost Database: test
```

```
-- -----
```

```
-- Server version 5.1.50-community
```

```

/*!40101                                                    SET
@OLD_CHARACTER_SET_CLIENT=@ @CHARACTER_SET_CLIENT */;

/*!40101                                                    SET
@OLD_CHARACTER_SET_RESULTS=@ @CHARACTER_SET_RESULTS
*/;

/*!40101                                                    SET
@OLD_COLLATION_CONNECTION=@ @COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@ @TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014      SET      @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014                                                    SET
@OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101      SET      @OLD_SQL_MODE=@ @SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@ @SQL_NOTES, SQL_NOTES=0 */;

--

-- Table structure for table `roles`

```

--

DROP TABLE IF EXISTS `roles`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `roles` (

 `id` int(3) NOT NULL AUTO_INCREMENT,

 `nombre` varchar(100) NOT NULL,

 `descripcion` varchar(100) DEFAULT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--

-- Dumping data for table `roles`

--

LOCK TABLES `roles` WRITE;

/*!40000 ALTER TABLE `roles` DISABLE KEYS */;

```
INSERT INTO `roles` VALUES (1,'Usuario Externo','Usuario Externo'),(2,'Usuario Interno','Usuario Interno');
```

```
/*!40000 ALTER TABLE `roles` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```

```
/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
```

```
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
```

```
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

```
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

```
-- Dump completed on 2011-11-21 17:27:37
```

```
CREATE DATABASE IF NOT EXISTS `test` /*!40100 DEFAULT CHARACTER SET latin1 */;
```

```

USE `test`;

-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)

--

-- Host: localhost Database: test

-- -----

-- Server version 5.1.50-community


/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;

/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;

/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014 SET
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

```

```
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
--
```

```
-- Table structure for table `roles_usuario`
```

```
--
```

```
DROP TABLE IF EXISTS `roles_usuario`;
```

```
/*!40101 SET @saved_cs_client = @@character_set_client */;
```

```
/*!40101 SET character_set_client = utf8 */;
```

```
CREATE TABLE `roles_usuario` (
```

```
  `id_rol` int(3) NOT NULL,
```

```
  `id_usuario` varchar(20) NOT NULL,
```

```
  PRIMARY KEY (`id_rol`,`id_usuario`),
```

```
  KEY `id_usuario` (`id_usuario`),
```

```
  KEY `FKDD371BAC653E8806` (`id_usuario`),
```

```
  KEY `FKDD371BACFE4041D1` (`id_rol`),
```

```
  CONSTRAINT `FKDD371BAC653E8806` FOREIGN KEY (`id_usuario`)  
REFERENCES `usuarios` (`id`),
```

```
CONSTRAINT `FKDD371BACFE4041D1` FOREIGN KEY (`id_rol`)
REFERENCES `roles` (`id`),
```

```
CONSTRAINT `roles_usuario_ibfk_1` FOREIGN KEY (`id_rol`)
REFERENCES `roles` (`id`),
```

```
CONSTRAINT `roles_usuario_ibfk_2` FOREIGN KEY (`id_usuario`)
REFERENCES `usuarios` (`id`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
/*!40101 SET character_set_client = @saved_cs_client */;
```

```
--
```

```
-- Dumping data for table `roles_usuario`
```

```
--
```

```
LOCK TABLES `roles_usuario` WRITE;
```

```
/*!40000 ALTER TABLE `roles_usuario` DISABLE KEYS */;
```

```
INSERT INTO `roles_usuario` VALUES (1,'80175381'),(2,'52973571');
```

```
/*!40000 ALTER TABLE `roles_usuario` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;
```



```

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
*/;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101                                     SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101                                     SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101                                     SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2011-11-21 17:27:37

CREATE DATABASE IF NOT EXISTS `test` /*!40100 DEFAULT
CHARACTER SET latin1 */;

USE `test`;

-- MySQL dump 10.13 Distrib 5.1.40, for Win32 (ia32)

--

-- Host: localhost Database: test

-- -----

-- Server version 5.1.50-community

```

```

/*!40101                                                    SET
@OLD_CHARACTER_SET_CLIENT=@ @CHARACTER_SET_CLIENT */;

/*!40101                                                    SET
@OLD_CHARACTER_SET_RESULTS=@ @CHARACTER_SET_RESULTS
*/;

/*!40101                                                    SET
@OLD_COLLATION_CONNECTION=@ @COLLATION_CONNECTION */;

/*!40101 SET NAMES utf8 */;

/*!40103 SET @OLD_TIME_ZONE=@ @TIME_ZONE */;

/*!40103 SET TIME_ZONE='+00:00' */;

/*!40014      SET      @OLD_UNIQUE_CHECKS=@ @UNIQUE_CHECKS,
UNIQUE_CHECKS=0 */;

/*!40014                                                    SET
@OLD_FOREIGN_KEY_CHECKS=@ @FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0 */;

/*!40101      SET      @OLD_SQL_MODE=@ @SQL_MODE,
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

/*!40111 SET @OLD_SQL_NOTES=@ @SQL_NOTES, SQL_NOTES=0 */;

--

-- Table structure for table `usuarios`

```

--

DROP TABLE IF EXISTS `usuarios`;

/*!40101 SET @saved_cs_client = @@character_set_client */;

/*!40101 SET character_set_client = utf8 */;

CREATE TABLE `usuarios` (

 `id` varchar(20) NOT NULL,

 `nombre` varchar(100) NOT NULL,

 `apellido` varchar(100) NOT NULL,

 `usuario` varchar(20) NOT NULL,

 `password` varchar(20) NOT NULL,

 `email` varchar(100) NOT NULL,

 PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40101 SET character_set_client = @saved_cs_client */;

--

-- Dumping data for table `usuarios`

--

```

LOCK TABLES `usuarios` WRITE;

/*!40000 ALTER TABLE `usuarios` DISABLE KEYS */;

INSERT INTO `usuarios` VALUES
('52973571','Andrea','Ravagli','cravagli','098765','cravagli@gmail.com'),('8017
5381','Julian','Osorio
Amaya','julian.osorio','123456','julian.osorio@gmail.com');

/*!40000 ALTER TABLE `usuarios` ENABLE KEYS */;

UNLOCK TABLES;

/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
*/;

/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

```

